

BMUG •

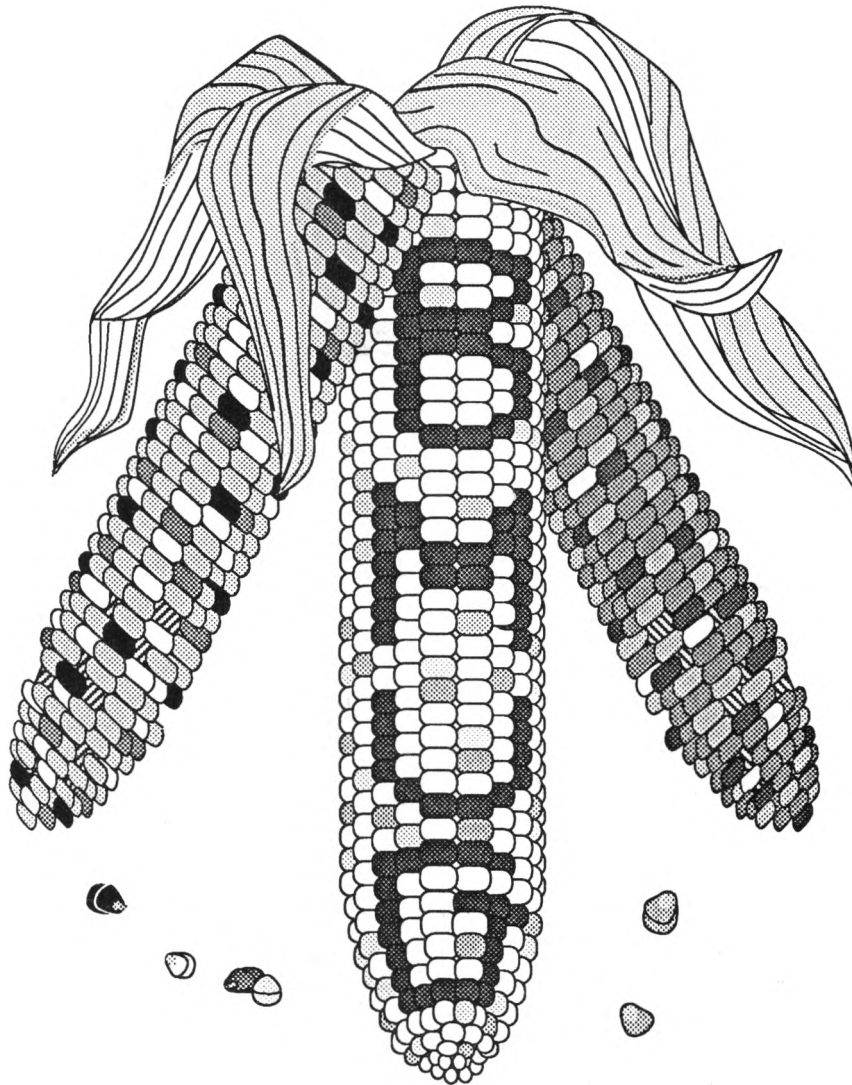
L
A
T
E

1
9
8
6



The Fall 1986

BMUG Newsletter



Announcement Line: (415) 849-9114
Help Hot-Line: (415) 849-HELP
BMUG BBS WEST: (415) 849-BMUG
BMUG Business Line: (415) 549-BMUG
MCI Mail: BMUG

CompuServe: 70007,2271
Delphi: BMUG
GErie: BMUG
The WELL: BMUG
AppleLink: UG001

Copyright © 1986 BMUG, Inc.
1442A Walnut St., Suite 62, Berkeley, CA 94709

BMUG, Inc. is a California Educational Nonprofit Corporation, and a registered student group at the University of California at Berkeley. BMUG membership is \$40.00 per year (including domestic postage and handling). It includes regular publications containing meeting notes, news, product reviews, product recommendations and reference information, accompanied by a disk containing selected choice public domain software; access to the bulletin board system and telephone help hot line; and access to the regular General Meeting and Special Interest Groups. For more information see the back of this newsletter.

The material appearing in this newsletter may not be reproduced, in whole or in part, for any commercial purpose without written permission from BMUG, Inc. Articles in the newsletter which are not specifically copyrighted by the author may be reproduced for non-profit distribution between individuals or within other non-profit users groups, **if and only if**, proper citation is made to "BMUG, Inc., Berkeley, California, USA," and the BMUG newsletter issue in which the material appeared. Copies of any such reproductions must be sent to BMUG. Opinions expressed herein do not necessarily represent those of BMUG or the University of California or its affiliates. Throughout the newsletter, titles of trademarked software and hardware products are presented. All such proper names are assumed to be trademarks of their manufacturing companies.

Cover leaf by Dan Wood

Title page corn by Carol Morita

Page header graphics by Kim Payne

Additional artwork and design by Steve Costa, John Heckendorn, Beverley Kane, and Scott Kronick

Back cover illustration adapted by Dan Wood from Berke Breathed's original, used with permission from the Washington Post Co.



Table of Contents

INTRODUCTIONS

Introduction	Reese Jones	1
• The state of the group address		
The "Newsletter"	Kee Nethery	8
• What we do in between ice cream runs		

DESIGN, GRAPHICS & ANIMATION

Beyond Pretty Pages	Ann Casamajor	11
• Planning and assembling a booklet		
Zen and the Art of the Macintosh	Frank Patrick	14
• A book for those without a motorcycle		
Tutorial: Silicon Press	Barbara J. Chan	16
• Printing cards, labels, and leaflets, plus color		
SuperPaint: Its Name is True	Dan Wood	20
• Paint and draw in one program		
A Preview of Cricket Draw	Dan Wood	23
• Producing modifiable PostScript files directly		
ComicWorks & GraphicWorks	Dan Wood	25
• The page layout program with two names		
Towards a Strategy for Macintosh Explosions	Glenn Harris	27
• Improving your destructive capability		
High Speed Animation on the Mac	Philip J. Suh	29
• A powerful way to produce animation		
How 3D Graphics is Done on the Mac	Greg Simons	31
• What's inside three 3D programs		
Programming in PostScript	Denis G. Pelli	35
• New graphic capabilities through coding		
Adventures with PostScript	Arthur Luehrmann	45

BITS & BYTES

File and Disk Recovery for Just About Anyone • You should have made a backup, but nooooo...	Steve Costa	57
A Picture of Everything a Computer Knows • A glimpse of assembly language for first timers	Scott Kronick	65
Using Font/DA Mover • Installing and removing fonts, DAs, and more	Stephen Howard	70
Editing Your DSAT • A ResEdit challenge	Christian Wiedmann	73
ResEdit: Hacking for the Rest of Us • Changing menus, cursors, dialogs, and icons	Christian Wiedmann	75
The Animator: A Simple Lightspeed Pascal Program • A programming exercise using animation	Linda Custer	78
ZBasic • A quick review by a BASIC user	Ken Hule	85
Bypassing the Trap Dispatcher for Fun and Profit • A hackers tool for speeding up applications	James E. Hopper	88

PRODUCTIVITY TOOLS

File Finders • Three programs that locate files	Mark Loncar	93
Koala KAT-ing Around the Classroom • A mouse alternative	Kenneth William Hirsch	96
Micrograde and the Joys of Teaching Classes • Grade bookkeeping and calculations	Jasper Rine	98
All the Home Accounting Programs • Five programs reviewed	Scott Beamer	101
Mini Reviews • On time, tax, and disk zapping	Scott Beamer	107
Smart Alarms • An appointment DA	Steve Cisler	110

WORD PROCESSING

WriteNow and HabaWord • The second generation of Mac word processing	John Heckendorn	113
The New Word Processors • More word processing alternatives	Peter Korn	121
Word Handler • Yet another view	Kenneth William Hirsch	126

EgWord	Paul Swingle	128
• A Japanese word processor		
More Than You Want to Know About Spell Checkers	Scott Beamer	132
• A comparison of eight programs		
Fonts	Richard S. Palais	145
• An introduction to font design		
Editors	Gordon Haig	151
• Two editors to use along with MacWrite		
Document Compare	Kenneth William Hirsch	154
• A program to compare text files		
WYSIWYG Technical Word Processing	Richard S. Palais	159
• The current state of equation processors		
An Equation Maker for the Mac	Doug Shier	164
• A DA for including equations in your documents		
T _E X for the Macintosh from K&S and FTL	Richard S. Palais	167
• Two very different implementations of T _E X		
FeiMa Chinese Language Processor	Barbara J. Chan	169
• Chinese character processing	illust. Chris Molé	
ARTIFICIAL INTELLIGENCE		
Electricity for My Dreams	Beverley Kane	175
• An Overview of Artificial Intelligence		
Prologue to LISP	Beverley Kane	180
• LISP compared with Pascal and C		
Experlisp 1.5	Bill Woodruff	184
• An advanced look at the classic AI language		
MacScheme	Andrew Shalit	189
• A dialect of LISP reviewed		
ExperProlog II	Dan Shafer	192
• The AI language newcomer		
Robotics	Jim Strobe	199
• Making machines on the Mac		
Alfonso's Head is Half-Connected	Alfonso Qwerty	202
• An interview with Racter		
Alice's LISP Machine	C. Stacy, A. Wechsler, &	205
• A saga from MIT	N. Chiappa	
BMUG Choice AI Products	AI Staff	210
• Fact and fantasy		

PRINTING

Practical Tips for LaserWriter Authors • Getting a step on trial and error	Ronald A. Schmidt	213
Tips on the LaserWriter • And Other Postscript Printers and Typesetters	Arthur Naiman	218
Linotronic L100 Tips • Lots more dots per inch	Ronald A. Schmidt	235
Using the HP THINKJET and QUIETJET+ • Hewlett-Packard printers	Kenneth William Hirsch	236

HARDWARE

Speeding up the Old Mac 512 • When the watch seems to stop	Joseph Schaller	239
Why Everyone Should Get the New ROMs • The decision to upgrade	Gordon Haig	246
I'm Upgrading! • More reasons to beef up the Mac	Ken Wait	248
The heart of the Apple II is still beating • About the Apple IIGS	Linda Custer	249
Radio Shack Model 100 Portable Computer • Making it a Macintosh add-on	Gordon Haig	253
A Day at the Races with MacCharlie • In the IBM den	Vicken Khachadourian	263
Keyboard Wrist Rest • A do it yourself project for tired typists	Kenneth William Hirsch	266
Locking your Macintosh • Alternate security kits	Cliff Stoll	268
Hush, Micah! • Installing an on-off switch for a hard drive	Michael Lamoureux	269
Tips for Hard Disk Users • Getting the most out of your hard drive	Dan Callaway	271
Building Your Own Hard Disk • Assembling a SCSI disk from scratch	Tim Standing	274
Making Printed Circuit Boards on a Macintosh • Using MacPaint to design and make PC boards	Bryan McDonald	288
SCSI Hacking • General purpose I/O with the SCSI port	Michael Lamoureux	294

Adjusting Your Mac	David W. Berry	300
• With a voltmeter and guts, you can...		
CONNECTIVITY		
AppleTalk Networking	Michael Rogers	303
• Large-scale corporate networking		
MACLINK-ing CP/M and MS DOS	Kenneth William Hirsch	310
• Wordstar: From IBM to Mac and back		
MacLink: IBM Data Transfer to Mac	Harry Critchfield	313
• Simple data transfer from an IBM PC to a Mac		
The Mac Underground	Chris Halaska	314
• Connecting to an on-line intelligence system		
CompuServe Update	Linda Custer	320
• Dungeons and Dragons On-Line		
2400 Baud Modems	Timothy Standing	322
• Experiences with faster modems		
Strategic Conquest Plus	Steven Fransen	323
• Connecting with a Mac game		
MCS: Multichannel Communications System	Paul B. Blood	326
• Transatlantic transmission of files using Macs		
Two for the Lab: MacADIOS and LabVIEW	Paul F. Daley	328
• The laboratory interface grows up		
BUSINESS		
You Don't Need \$250,000!	Shepard Tamler	335
• Publishing Mac software successfully		
A Review of the Excellent Exchange	Don Scellato	338
• A collection of Excel applications		
Microcomputer Consultanting	Audrey D. Cole	341
• Pointers and pitfalls in working for yourself		
Big Guns: Double Helix vs. Omnis 3 Plus	Rick Eisner	343
• Comparing the two database systems		
dBase Mac and dMac III	M. Joseph Schaller	346
• Comparing the Mac and IBM database systems		
Record Holder	David Morgenstern	349
• A simple and cheap database system		
Poor Tom	Glenn Harris	351
• An associative database		

Macintosh Statistics Software	Michael La Belle & Melissa Austin	353
• Six programs for statistical analysis		
COMMENTARY		
Computers in Berkeley?	Enrique Zambrano	361
• The Mac in local politics		
Australia and the Macintosh	Ronald A. Schmidt	364
• What's up down there		
Piracy and Rentals	Scott Beamer	366
• Legal update		
Just Say No to Piracy	Scott Beamer	369
• Nancy takes a stand		
Diary of a MacNeophyte	Glen Frendel	371
• How to be a free Beta test site		
How Do We Pick Our Computers	Enrique Zambrano	379
• Let me count the ways		
A Beginner's Guide to Getting Help	Mark R. Chandler	382
• Is there anything besides BMUG		
Herb Cain Punches John Dvoruk for Making Pass at Dear Abbey	Scott Kronick	383
• Ugly scene mars desktop publishing expo		
BMUG CHOICES		
BMUG Choice Products		389
• The very good, the good, or the only		
The Latest BMUG Disks		402
• Desktop views of recent library additions		
BMUG Newsletter Disk #60		403
• Info on the disk that comes with this newsletter		
BMUG Membership and Product Order Form		405
• Make copies and give to friends		

Table of Specials

Illustrations

San Francisco Skyline	Jones	7
BMUG Meeting Attendees	Kronick	7
Circle	Chan	19
Dark Castle	Silicon Beach	30
Architectural Drawing	Kanemoto	56
Big Eyes	Callway	77
Boxes	Hockabout	92
Puffs	Wood	97
Ballons	Wood	100
SpiderWeb	Wood	106
Watch	Wood	109
Shoe	Wood	111
Fun and Games	Wood	112
I-House	Chan	120
Star Scream	Zambrano	144
Processing Words	Chan	166
Boxes	Hockabout	168
Flying Horse	Molé	169
Arabic Calligraphy	Godlas	173
Holiday Graphics	Kanemoto	174
Dark Castle	Silicon Beach	198
Pen	Wood	234

Race	Hockabout	252
Peruvian Lily	Chan	264
Steve Wozniak	Kane	265
Scroll	Wood	299
TV	Wood	301
Faces	Wood	302
Blocks	Hockabout	309
Start-up screens	Suh	334
Boxes	Hockabout	337
Q-Graphics	Cole	345
I-House II	Chan	348
Flower Cat	SunShine Graphics	370
Bill the Cat Scan	Suh	381
Glass Wire	Wood	387
Cappuccino Mac	Howard	388
 BBS Excerpts		
Mac Draft 1.2 Tips		19
Grateful Dead Request		69
Pascal Comments		83
Pascal Comments		84
Mac-Chimney		125
Large Screens?		158
Mac Virtues		183
LaserBoard BBS		188
Apple // in a Mac		217
Bmug Graphic and Design SIG		267
Micah Question		268

Mac Hard Disk Drives	270
Make a Mac Hard Disk Drive	293
HP 12C DA	313
Big Mac Screen	322
MS Word Question	327
Mac Virtues	340
Avatex Modem	342
Macs in Tolman	352
Micah Woes	401
Tips	
MacLabeler	64
Where to get SCSI connectors	363

BMUG
Fall 1986



Introductions

INTRODUCTIONS

Introduction

- The state of the group address

Reese Jones

1

The "Newsletter"

- What we do in between ice cream runs

Kee Nethery

8

BMUG
Fall 1986



An Introduction to BMUG

Reese Jones

BMUG, Incorporated (previously known as the Berkeley Macintosh Users Group) is located at the University of California at Berkeley and is a registered student group. This volunteer run group is primarily devoted to the exchange of information about microcomputers, primarily the Macintosh and related products. Also supported are the new generation of UNIX based university workstations including the Sun 3, DEC MicroVAX, IBM RT and others. A particular focus of BMUG is the use of personal computers in the academic environment and the interconnection of different brands of computers through networks and telecommunications.

BMUG's membership consists of roughly 3000 people: students, faculty, and staff of U.C. Berkeley, Lawrence Berkeley and Livermore National Labs, numerous other universities and research institutions, as well as a substantial number of members from the surrounding community and all over the world.

BMUG, Incorporated (A California Educational NonProfit Corporation)

BMUG Inc. became legally incorporated in December 1985. BMUG Inc. is now a California educational non-profit corporation and has received 501c3 tax exempt status, which makes any contributions to BMUG tax deductible.

BMUG is volunteer-run and is budgeted to be self funding and hopefully break even. BMUG has a paid support staff to handle operational tasks while most BMUG activities are a result of the efforts of numerous enthusiastic and creative volunteers. Prices are set to meet cost of materials and to cover operational expenses for the various services provided by BMUG like the office staff, phone help lines, BBS systems, equipment testing, meeting and mailing costs, etc. Approximately 90% of BMUG's budget goes directly towards cost of goods and direct costs of servicing the stated purpose of the group. BMUG is entirely self-funded and receives no funding from Apple Computer Inc. or the University of California. BMUG takes no advertising (a substantial financial concession) in order to maintain as objective a view as possible. BMUG functions primarily on the substantial efforts and good will of its membership.

BMUG People Include:

Kee Nethery	Beverley Kane	Scott Kronick	Steve Costa
Raines Cohen	Linda Custer	Barbara Chan	Harry Critchfield
John Heckendorn	Kent Hockabout	Tony Vignaux	Gordon Haig
Michael Rogers	Lori Brooks	Anne Levine	Phil Suh
Reese Jones	Tenley Sutton	Amy Roberts	Lisa Cort
Michael Lamoureux	Sam Roberts	Fred Huxham	Tom Santos
Dan Wood	Bill Woodruff	Kelley Wicks	Jay James
Ann Casamajor	Carl Di Giorgio	Bernard Aboba	Rex Dog

Help Lines

The BMUG help line is usually staffed during the afternoon/evening hours on weekdays. Call **(415) 849-9114** for announcements or to leave a message (we will try to return your calls) and **(415) 849-HELP** to talk to a person (usually the infamous Steve Costa, or leave a message and we will try to return your call). To reach the BMUG office call **(415) 549-BMUG** during business hours. By modem call **(415) 849-BMUG** to reach the BMUG BBS. On the help lines we try to



BMUG FALL 1986

answer your questions or refer you to someone who can. Members who are familiar with a particular program or topic have volunteered to answer Macintosh questions over the phone during a particular time and day (such as Sunday 4 to 5 pm). If you can help answer questions on a particular topic or program, please volunteer; it can force you to really learn your topic.

The BMUG Newsletter

Minutes from the weekly meetings are published regularly on the BMUG BBS with excerpts reprinted in the BMUG Newsletter which is published twice a year and also contains news, product reviews, product recommendations, tutorials, and reference information. The BMUG newsletter does not include advertisements because we wish to maintain a high level of objectivity. Neither BMUG Inc. nor the University of California is responsible for any damage or loss resulting from the information, software or hardware products distributed by BMUG.

The newsletter consists of material submitted by the membership. See Kee Nethery's article in this section for information on how to contribute. Submissions don't have to be long. Contributors will be properly credited and articles may be copyrighted by the author. **Let's share what we've found out.** The Spring 1987 newsletter is already underway!

General Meetings

BMUG holds at least 3 meetings every week. BMUG general meetings are held every Thursday at 6:00 PM in PSL (Physical Sciences Lecture Hall) on the U.C. Campus, near the East Gate of the campus. (In the event we can't reserve PSL for some reason, the new location will be posted there.) Meetings consist of new product demonstrations, tutorials, and/or presentations by invited guests, following the first hour of open discussion, demonstrations, and question-and-answer sessions. Attendance is typically between 250 and 350 people. We try to keep the atmosphere informal, particularly towards the end of the meeting when many people linger to exchange ideas or look more closely at the demonstration equipment. The room is well-equipped for the meetings with a HUGE video projector (a Hughes high resolution projector directly connected to a Mac) for viewing the Mac's or other computers' window during demonstration, or video tapes. Meetings are always free and open to the public; you need not be a member to attend meetings or to purchase BMUG distribution disks (\$3 each), blank disks, or other BMUG products. New members may register before or after the meetings. We also distribute materials and handouts from software and hardware vendors at the meetings.

Beginners' Group Meetings

The BMUG Beginners' SIG (Special Interest Group) currently meets every Tuesday at 6:00 pm (call the announcement line to find out the location). The meetings are small and informal, and are tailored to the new user who may be lost among the often technical presentations and gossip at the general meetings.

Graphics/Publishing SIG Meetings

The BMUG Graphics SIG currently meets every Monday at 8:00 pm at Byting Your Time computer parlor on Durant below Telegraph (call the announcement line to make sure). The meetings are small and informal, and oriented to those who are particularly interested in graphics (printed or screen), desktop publishing, PostScript programming, digitizers, typesetters etc.

Beginning Developers' Special Interest Group

The BMUG Beginning Developers' SIG currently meets alternate Wednesdays at 6:00 PM in 30 Wheeler on the U.C. campus during the school year. (Call BMUG to verify meeting and location before attending.) The Beginning Developers' Group is oriented to those who are starting out

BMUG Fall 1986



programming stand-alone applications on the Macintosh and deals with the basics of creating programs in MPW, Lightspeed Pascal and the like.

Advanced Developers' Special Interest Group

The BMUG "Advanced" Developers' SIG currently meets on the other alternate Wednesdays (vs Beginning Developers Group) at 6:00 PM in 30 Wheeler on the U.C. campus during the school year. (Call BMUG to verify the location before attending.) The Developers' Group Spring 1985 Newsletter is available to BMUG members for \$18 each. It contains tutorials and examples of programs in Consulair C from the Spring 1985 meetings. BMUG disk #18, included with the newsletter, contains source code and compiled programs that supplement the Developers' Newsletter. The Advanced Developers' Group is oriented to those who want to discuss the subtleties of programming on the Macintosh and deals primarily with MPW and Consulair C.

Computer Music Users Group (CMUG)

The BMUG music SIG meets approximately every 3 weeks to discuss the use of the computers in music. There are demonstrations, discussion on such topics as MIDI control and composition. CMUG usually meets on Saturday at Leo's Music on Telegraph at 57th in Oakland. Call the announcement line for details on the next meeting.

Educational Software SIG

The Educational Software SIG meets now and again and deals with issues in the development of course ware for the university environment. The group tries to bring together instructors, developers and others interested in the uses of computers in education.

Trade Shows and Festivals

BMUG operates information booths at most of the major computer trade shows, and in the fall of each year, BMUG hosts the Berkeley MacToberFest. Upcoming shows of particular note include the **MacWorld Expo/San Francisco** (Jan.8-10th), the original Macintosh exclusive computer show, to be held at the **Moscone Auditorium**, San Francisco. The MacWorld shows are usually well run and are particularly worth going to, with hundreds of different companies showing their products there, often things you can't find in stores. There are also seminars and demonstrations to see. Contact Mitch Hall Associates (617) 329-7466 for more information. Come help at the BMUG Booth or stop by and visit when you are there.

Bulletin Board Systems

BMUG currently operates a BBS system in Berkeley running 24 hours a day, seven days a week. We are currently running on Scott Watson's Red Ryder Host software. To use the BBS, dial **(415) 849-BMUG** (that's 849-2684 for you digital people) at 2400/1200/300 baud. You don't need an account to log on for the first time, the BBS will prompt you on what to do (leave a password of your choice the first time). Then until your account validated, you will be given five minutes per call to explore the system and to write to the SYSOP asking for validation. Raines Cohen is BMUG's Berkeley BBS principal SYSOP. The SYSOP will check your membership and try to validate you within a couple of days (remember SYSOPs take vacations too.)

When you are validated, you will have full access to mailboxes, news (including a rumors section hosted by Oliver Wendell Jones III, featuring the *juiciest* computer gossip you'll ever see), and chatting with a SYSOP (if he/she is around and awake). You can upload submissions (new software, newsletter submissions, etc.) or download files from our massive library, which has over 20 megabytes online.

If you try calling and the line is busy, don't despair! BMUG BBS's are understandably popular



BMUG FALL 1986

and frequently in use (3-6am PST is the best time to get through). We're looking into expanding into a multiple-line system; call or write if you can help.

BMUG and the online services

BMUG also follows or maintains a section and a presence on most of the major national online network services including CompuServe, Delphi, GENie, DARPAnet/Usenet, The WELL and others. If you can help with any of these, please call.

MacRecorderII™ Kit

BMUG offers a kit called MacRecorder II, which is a speech digitizer and analog to digital converter (hardware and software combination). Cost is \$55 for the kit, which includes the circuit board, all parts, box, power supply, microphone, complete software (source code is also available), assembly instructions for building the device, and shipping. See the articles by Michael Lamoureux in previous issues for more information on the design of the MRII. Assembled/tested kits are available from Farallon Computing Berkeley (415)849-2331. MacRecorder II is completely hardware compatible with the MacNIFTY sound digitizer. The MRII has applications for laboratory A/D data collection, recording voice/music on the Macintosh, real time oscilloscope display, displaying and analyzing wave forms on the screen, incorporating recorded sounds into your programs, speech recognition, alarm systems, music and who knows what; give your Mac ears!

Adaptor Cables and other Hardware

BMUG distributes a variety of difficult to find adaptor cables (Plus to DB-9, modem cables, keyboard extension cables, etc), hardware kits (MacRecorderII) and group purchase items (modems, mousepads, blank disks etc.). These are often difficult to find or expensive in a store, so we get the parts in quantity and can thus sell them for approximately half of the usual retail cost. See the hardware notes (here and in back issues) for information on making your own cables or adaptors for various purposes. Also note we periodically have "*Make your own cable nights*" or "*Install Your own SCSI port nights*" at BMUG meetings where we bring the parts and tools then help you solder them yourself. BMUG, however is not in the business of selling computer supplies but distributes these items for their educational value.

BMUG Video Tape Library

BMUG maintains a video tape library of significant meeting presentations, interesting lectures, computer related TV shows, and Macintosh advertisement and promotional tapes. 4-hr. VHS tapes are available for \$20 from BMUG. Call or write the office for information.

BMUG Products Distribution

BMUG products are available at Thursday BMUG meetings or by mail at 1442A Walnut St.#62, Berkeley CA 94709. BMUG products are also available at a couple of local *computer parlors* : **Copy-Rite** copy center (415)-548-4433 at 2154 University Ave., (Between Shattuck and Oxford, open till midnight) and at **Winner's Circle** Computer Dealer Telegraph Ave, near Derby.

Referrals

We often get calls from companies and individuals looking for freelance programmers, consultants, or instructors to work on short- or long-term projects involving the Mac or to teach classes about the Mac. We maintain referral lists, both of openings and of available freelance specialists from our group. If you are a company or individual looking for a consultant, programmer, or instructor/tutor, or if you are a Mac programmer, instructor, or freelancer who is looking for a project, call the BMUG HELP line so that we may connect you with the appropriate company or person.

BMUG
Fall 1986



Floppy Disk Review Project

BMUG is currently conducting a large scale floppy disk test project where we are distributing 1,000 of each major brand of 3.5 inch floppy disks in the market. The disks are labeled on the back with their batch number and the return address and phone number to report any failures. The disks are passed out (for free) at shows, meetings and other computer events with the instructions to **"use the disk in your daily routine, and if it fails for any reason to return it for a replacement."** The disks are **brand-new** off-the-shelf units from the manufacturers' normal production runs and have **NOTHING ON THEM** (some confusion there) and, if you can format the disk in your Mac, they should work as well as any other new disk (SS). So please use the disks for whatever and report back if you have any problems.

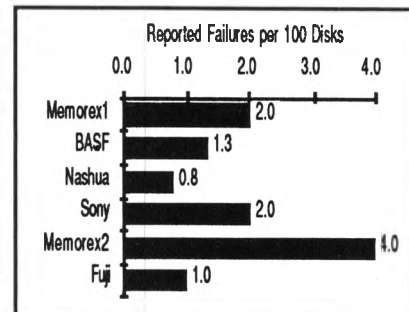
So far we have passed out 6000 between February and October and we are tracking the failure rate of these disks and the remaining brands of disks to be distributed over the next year. We are keeping statistics on initial failure rates, and how long the media and cases last under normal use for each brand. Ten of each brand's one thousand disks are damaged by us before distribution (Controls) so we can also keep proper reporting statistics for disks distributed at different events (see Tables). Objectively evaluating the quality of different brands of floppy disks is a very difficult task because failure rates are typically on the order of 1 in 100; thus to get adequate statistics, a large number of disks must be tested in a carefully controlled protocol. We are most interested how the disks perform in normal use and have designed this test to evaluate that.

We have received back several failed disks and have compiled here some preliminary statistics on the test results. These results should be viewed cautiously because they are preliminary and include only the **user reported failures**, we are currently evaluating and re-testing each of the returned disks to see if they cannot be formatted in our machines and why. The statistics are interesting nonetheless, particularly in that all of the brands had roughly similar failure rates (in the range of 1 in 100) compared with our previous experience a year ago where some brands seemed to have failure rates as much as 5 times higher (1 in 20). We were somewhat disappointed with our control reporting rates, 10-40%, indicating that **some of you have not been returning your failed test disks**. Please do so; the results will improve our statistics and aid us all in evaluating these different brands of disks. We intend to **follow the lives of these and several other brands of disks over the next year** to report on media failure rates (in contrast to initialization failures reported here).

Some Preliminary Statistics from the Disk Brand Test

Disk Brand	Color	Distribution Time Period	Total # of Disks Distributed	Number of Control Disks	# of Controls Returned	% Controls Reporting	Total # of all Disks Returned (all disks)	Total # of Test Disks Returned (non control)	User Stated Reason for Return			
									none	Init	Use	Confused
Memorex	Blue	Jan/Feb 86	1000	10	1 of 10	10%	3	2	0	2	0	0
BASF	Tan	Jan/Feb 86	1000	10	3 of 10	30%	11	8	1	2	1	4
Nashua	Black	Jan/Feb 86	1000	10	4 of 10	40%	7	3	0	1	2	0
Sony	Blue	Mar/Apr 86	1000	10	4 of 10	40%	12	8	4	2	2	0
Memorex	Blue	Sept/Oct 86	1000	10	1 of 10	10%	6	5	2	1	1	1
Fuji	Blue	Sept/Oct 86	1000	10	1 of 10	10%	2	1	0	1	0	0

Disk Brand	User Stated Reason for Return			% of Cases where user was just Confused (thought disk had data)	% Reported Failure Type		
	none	Use	Confused		Init Fails	Use Fails	No Reason
Memorex	0	0	0	0%	100%	0%	0%
BASF	1	1	4	50%	25%	13%	13%
Nashua	0	2	0	0%	33%	67%	0%
Sony	4	2	0	0%	25%	25%	50%
Memorex	2	1	1	20%	20%	20%	40%
Fuji	0	0	0	0%	100%	0%	0%

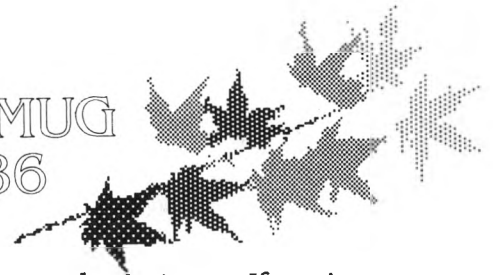


Disk Brand	Percent of all Disks Returned (including Controls)	Percent of Disks Returned (non Controls)	Percent of Test Disks Returned (w/Confusion Correction)	Control Reporting Rate	Calculated Overall Reporting Failure Rate Before Double Check (Corrected for % Controls Reporting & Confusion)	Expected Failure per 100 disks
Memorex	0.30%	0.20%	0.20%	10%	2.0%	2 ± 3 Failers per 100 Disks
BASF	1.10%	0.80%	0.40%	30%	1.3%	1 ± 2 Failers per 100 Disks
Nashua	0.70%	0.30%	0.30%	40%	0.8%	1 ± 2 Failers per 100 Disks
Sony	1.20%	0.80%	0.80%	40%	2.0%	2 ± 2 Failers per 100 Disks
Memorex	0.60%	0.50%	0.40%	10%	4.0%	4 ± 3 Failers per 100 Disks
Fuji	0.20%	0.10%	0.10%	10%	1.0%	1 ± 3 Failers per 100 Disks

Public Domain Software

BMUG has the most up-to-date public domain software library around. Disks are currently available to members for \$3 each (SS or DS!). See our detailed library listings at the back. We always offer the latest versions available, and we are constantly updating our software library.

BMUG Fall 1986



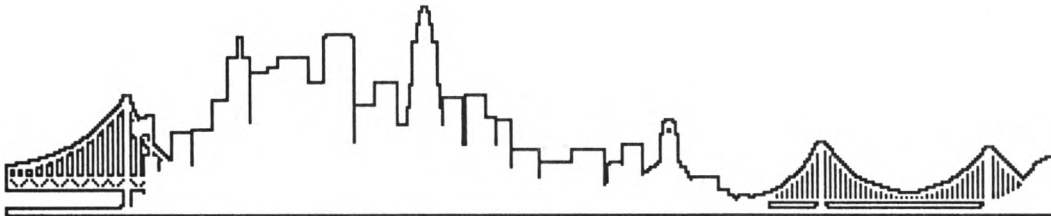
If you have any new software for the Macintosh, we welcome your submissions. If you've discovered a wonderful public domain or shareware program, have written one yourself, have created a wonderful template, or have drawn some interesting art, upload it to the BBS or send BMUG a disk. If you send us a disk, we will return a disk to you full of goodies! The maintenance of a comprehensive BMUG software library available to all depends on you!

Newsletter Back Issues

You can purchase back issues of BMUG newsletters. Each of the back newsletter issues come with a disk and can be purchased for \$15.

Newsletter Exchange Policy

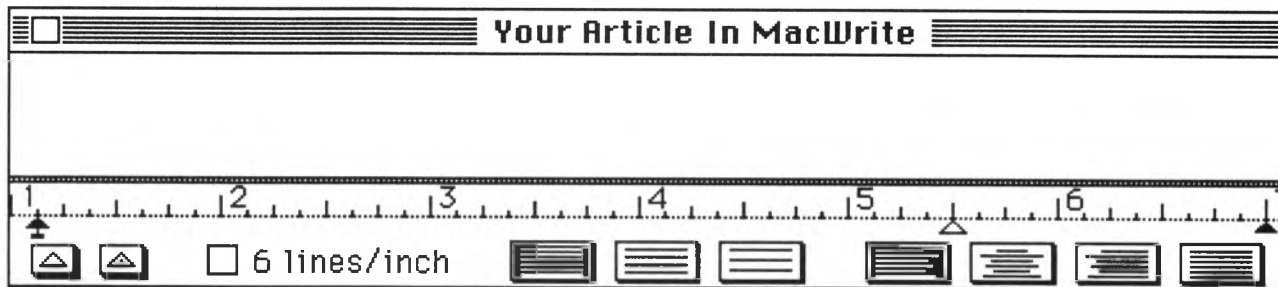
We now welcome exchanges with other users groups. Send your newsletter to BMUG, Attn: Steve Costa, at the address on the inside cover. Don't bother sending "zero-effort" newsletters or collections of meeting announcements. If you've put in some effort, we'll send back a newsletter in exchange. We intend to bind, index, read, and make newsletters sent to us available in a public location and we hope that you will do the same. We apologize to groups that, in the past, have sent us newsletters without ever hearing back.



SF Skyline / Jones



BMUG Meeting Attendees / Kronick



The “Newsletter” (Bookman, 18 Point, Plain Text)
Kee Nethery (Bookman, 14 Point, Plain Text)



Welcome to the BMUG "newsletter"! This Fall newsletter body is set in Bookman, 10 point, plain text, with no beginning paragraph indentation, at 6 lines per inch, single spaced, and with full justification. This article has several purposes: to describe what happens to make the newsletter come together, to show you how we want your articles formatted, and to request articles on specific topics for the next newsletter.

Building The BMUG "Newsletter": Bookman, 10 point, bold text

Each semester the newsletter *eventually* appears in your mail box. Two things are obvious: it's always late and each issue gets bigger. What's not obvious to you is what happens before it reaches you. A step-by-step article on how to build a newsletter would be nice for us to write, but every newsletter comes together differently. The newsletter team job descriptions are fairly consistent between newsletters. In practice, the editing team shares all these jobs and titles, and works together to bring you this publication.

Contributors: People like **you** write about what they are doing. Beginners describe the basics that we've forgotten were difficult to understand. Hackers satisfy our curiosity about what really happens inside our computer. And the rest of you tell us the truth about things we would like to know but haven't had the time to learn. Thanks!

Editor-In-Chief: In theory, the editor-in-chief is concerned with all the tasks involved in producing the newsletter. It's the editor's job to make sure that everything gets done. In practice, the editor worries about as much as possible and everything else gets handled by others who worry about why something is not getting done.

Section Editors: Some newsletter sections are decided ahead of time (like the AI section) and a person assumes responsibility for getting and editing articles on that particular topic. More typically, we receive several articles that naturally would be grouped together under one general topic and a person assumes responsibility for editing that group of articles. Section Editors also merge all their articles into one final MacWrite document.

Editors: Many people spend many weekends copy-editing each article, checking for accuracy,

BMUG Fall 1986



spelling, format, etc. They do the bulk of the newsletter work.

Art Editor: Arranges for the front and back covers, headers, and artwork to accompany articles or to be used as filler. Our local Graphics Design SIG (Special Interest Group) has supplied much of our artwork.

Production Coordinator: Finds the least expensive print shop that will do a good quality job. Past issues have been photocopied from LaserWriter prints and bound by the copy shop. The cost is high but small incremental orders are possible. This issue is offset printed and Perfect Bound because it looks better and is less expensive in large quantities.

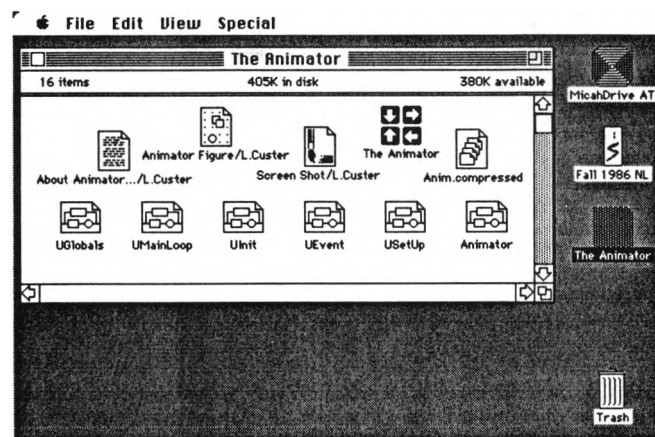
BMUG Office Staff: Unlike all these other jobs, entering your membership data into our database is a paid position because it's so important and needs to be done professionally. The staff, with occasional volunteer assistance, also handles packaging, postage, and addressing of your newsletter. When you're sending out thousands of newsletters, like we are, this is a huge job.

Software Librarian: Keeps our public domain and shareware software up-to-date, recovers newsletter files when they get trashed, and generally keeps up with what software has value. This person also ponders over what gets included in our membership disk.

On-Line Services: Several people spend large amounts of time monitoring the various on-line services (AppleLink, ARPAnet, BMUG BBS, CompuServe, Delphi, GENie, UseNet, MCI MAIL, etc.) They answer questions, collect information, and generally keep informed so that BMUG information is as up-to-date as possible.

Please Format Your Articles

We only accept articles on disk (in the past, we have accepted them by other means, including on-line transmission, but we have found some problems with the reliability of these methods). We like to receive them pre-formatted to our guidelines (just like this article). The text should be in *MacWrite*. If you don't have *MacWrite*, print out your article and give us a "text-only" file. I know, you love *MS Word*, *PageMaker*, etc. but we have to be able to pass your file around from person to person and we all own *MacWrite* or some word processor that can read *MacWrite* files.





Graphics can be included in your *MacWrite* document or you can send us *MacPaint* screen shots or *MacDraw* diagrams, or *GLUE* documents, or Startup Screens, or *FullPaint* or *SuperPaint* documents. We will handle putting the graphics into your article. Screen shots (MacPaint images of the current screen, like above) can be made by simultaneously pressing Command-Shift-3 in most programs.

The headers and footers need to be big enough to include our standard information. Please format them with four blank lines each, single spaced, at 6 lines per inch (check the box in the ruler).

Please keep the margins the same as what you get when you ask for a **New** *MacWrite* document. You can tab to your heart's content; please don't use spaces to make tables. Basically, look at how these articles are formatted and do what we do.

We try to adhere to certain style conventions. Menus, dialog choices, and other program selections, like **Save**, are **Bold** and capitalized. Keyboard commands are capitalized and hyphenated if they should be pressed simultaneously (Shift-Option-3). Software programs, and book and magazine titles, like *MacWorld* or *MacWrite* are Italic text and capitalized. Programming languages, like C or Pascal, hardware, fonts, DAs (Desk Accessories) and utilities are capitalized. We get a heavy dose of TLAs (Three Letter Acronyms)... Please spell them out at least once. If you review a product, include the company's Name, Address, Phone, Price, and if it's software, whether copy protected. Finally, in Berkeley, this editor is very careful to use non-sexist pronouns (instead of he or she). See why the editors have so much work to do?

We can send you a Writer's Disk for the next newsletter. The disk contains the screen versions of the laser fonts we will be using, a pre-formatted *MacWrite* document, and instructions. Send a note about what you will be writing about and include your return address.

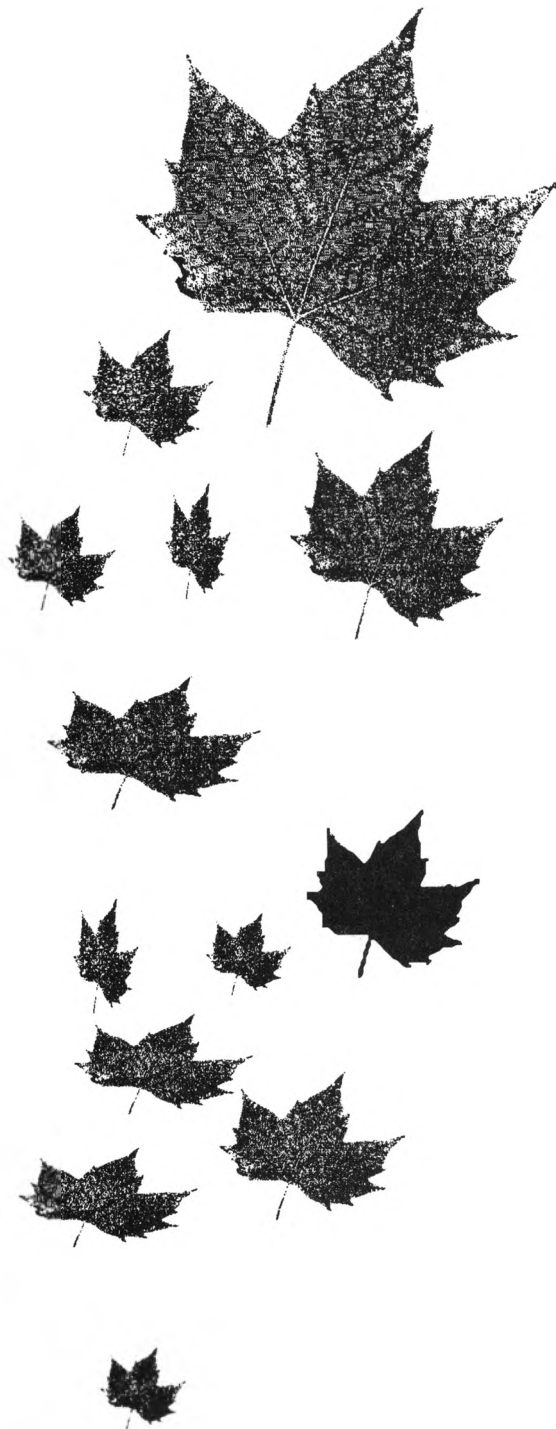
P.S. When you send us your article, write a small blurb about yourself. We (Mr. & Mrs. John Q. Public) are curious about you. So who are you??

Articles / Information We Would Like To Have For The Next Newsletter

- What's useful for the disabled?
- What's it like computing outside the United States? What should hardware and software designers include for markets outside the US? How do you keep up-to-date?
- Artificial Intelligence applications.
- Medical applications.
- How To ... articles (custom start-up screens and sounds, build your own hardware, etc.)
- Tricks and Tips for use as filler at the end of articles.
- Graphics and Publishing.
- Hardware peripherals (SCSI drives, optical character readers, digitizers)
- Networking (AppleTalk, Ethernet, corporate and university networks)
- Telecommunicating (running a BBS, on-line research, modems and mobile phones)
- Articles **you** would like to see.

Kee Nethery was the Editor-In-Chief for Spring 1986, and has been an absentee Editor-In-Chief for Fall 1986. His new car is fine, thank you, and he intends to keep it on the road. For grins he organizes large PhoneNet AppleTalk networks connecting various vendors together at Macintosh trade shows, Bookman, 10 point, Italics.

BMUG
Fall 1986



*Design, Graphics
& Animation*

DESIGN, GRAPHICS & ANIMATION

Beyond Pretty Pages • Planning and assembling a booklet	Ann Casamajor	11
Zen and the Art of the Macintosh • A book for those without a motorcycle	Frank Patrick	14
Tutorial: Silicon Press • Printing cards, labels, and leaflets, plus color	Barbara J. Chan	16
SuperPaint: Its Name is True • Paint and draw in one program	Dan Wood	20
A Preview of Cricket Draw • Producing modifiable PostScript files directly	Dan Wood	23
ComicWorks & GraphicWorks • The page layout program with two names	Dan Wood	25
Towards a Strategy for Macintosh Explosions • Improving your destructive capability	Glenn Harris	27
High Speed Animation on the Mac • A powerful way to produce animation	Philip J. Suh	29
How 3D Graphics is Done on the Mac • What's inside three 3D programs	Greg Simons	31
Programming in PostScript • New graphic capabilities through coding	Denis G. Pelli	35
Adventures with PostScript • Book production and fine art	Arthur Luehrmann	45



Beyond Pretty Pages: Planning and Assembling a Booklet

Ann Casamajor, J.D.

publish \vt 2 ...b: to produce or release for publication; *specif*: PRINT

OK, you've learned to use PageMaker or Microsoft Word and made yourself some pretty pages. Now what? Are you desktop publishing yet? That depends. A term paper or business report will generally go on 8-1/2 x 11 paper in a big stack just as it comes out of the LaserWriter. Velo-bind it and you're done. How many copies do you need? Step over to the photocopy machine and collate that many stacks before binding.

Sometimes, however, you might want your final product to be a different size. In my case, that size was dictated by the rules of the United States Supreme Court. After being shocked by the high cost of typeset, printed briefs in the peculiar size specified, I used my Mac to produce two different briefs that were needed in the course of drilling a dry hole in Washington, D.C. Our case was not among the three-in-a-hundred chosen for review by the court, but I was intrigued by the possibilities of publishing such briefs "on the desktop."

This article describes the process by which pages put together in Microsoft Word were assembled into a folded, stapled booklet approximately the size of a typical small software manual. Unless you're independently wealthy (or can otherwise justify the cost of a LaserWriter and a large-sheet photocopier), you'll probably need the help of your friendly local copy shop/laser parlor. I used Copy-Rite, partly because they are cozy with BMUG (selling BMUG disks and posting the help line number, among other things), and partly because their laser time is very reasonable. They're also friendly, helpful, and ready to take over whatever part of the process you may not want to handle yourself.

Supreme Court briefs must be 6-1/8 x 9-1/4, with a page image size of 4-1/8 x 7-1/8. I used Word, because it's what I'm used to and has some nice options, such as specifiable line spacing, which lend themselves to typesetting work and the minimalist graphics required to make an attractive cover/coverleaf page. A little fiddling with the margins, followed by some trial prints on the ImageWriter, set up the pages. I printed the final in 12-point Times, but used 10-point New York at home to approximate the same page size. Comparison to the expensive printshop product (\$800 and up) resulted in 15-point line spacing, which is set from the Word paragraph formats dialog box. This is slightly wider than single spacing and considerably more compact than double spacing. It's what the Big Boys use.

The two briefs were printed differently. The first was offset printed and the second was photocopied; there is no perceptible difference in the final product. I think I would offset only if extremely high quality was imperative, meaning the page proofs would come out of a Linotronic typesetter rather than the LaserWriter. Laser printers and photocopiers have approximately the same resolution, so little or no quality is lost in reproducing your pretty pages yourself, likely in the same shop in which you rented laser time.

Paste-up is required. I know of no software product that will allow you to lay out non-contiguous pages in landscape page orientations on 11 x 17 paper without nightmarish document-arrangement problems. Something might do it on a Linotronic, but I haven't had occasion to check it out. 11 x 17 is necessary because nothing in 8-1/2 width will be tall enough, when folded, to make the finished booklet. My finished unfolded page size was 12-1/8 x 9-1/4.



The briefs required a table of contents and a table of authorities cited, and also had a short appendix. The drill here is to print the text pages, beginning with Arabic page 1 (which may or may not carry a page number), consecutively from one document, then use the hard copy to peg the tabled locations in the text. Once such things as the tables and any other preliminary materials (preface, forward, author note, etc.) are arranged on pages in another document, number that group of pages with lower-case Roman numerals (i, ii, etc.) and print it out. You can also manually index this way. Open the document, use the search functions to find all references to key words and phrases, confirm the page location on your hard copy, and build an index document, with its own pagination.

The cover and coverleaf are essentially the same page, which contains the title of the court, case, and document as well as the names and addresses of the attorneys. The traditional typeset style is to put the case number at the top of the page, then a very tight pair of horizontal lines to bracket most of the page. A similar pair of lines at the bottom of the page completes the bracket. I used Times for everything, 18-point bold for the court's name, 14-point bold caps for the document title, sparing use of italics, and made the pairs of lines by making one (centered paragraph, hold down underscore) line, returning, doing another below it, selecting the two lines and choosing 2-point line spacing from the paragraph formats dialog box.. Experiment a little with length to balance its visual impact with the rest of the page, then copy it to the bottom.

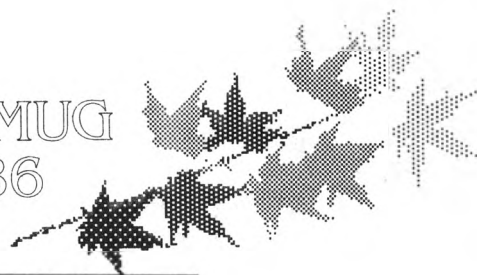
If you set your margins so as to leave the standard 3/4" at the top, paste-up is simplified. Slide the top of the printout along the top of the paste-up sheet until the justified margins are 3/4" from the edge of the paste-up sheet on the left and a mark made at 11-1/2" to the right. Real paste-up paper and a waxer are nice, but any clean white 11 x 17 will do, using a glue stick or rubber cement.

Before you can paste up, you must plan the layout of the folded sheets so the fronts and backs will all fall into the right order. This can fry your brain if you don't have a method and a good idea of how you want your booklet to look. First, make a map, or "plant," which is a pictorial diagram of the stack before folding and stapling. Use MacPaint or MacDraw (or something similar) to draw a bold horizontal line for the cover, then a series of thinner lines underneath. Label each face of the lines with the number of the page, including blanks for such things as every face but one of the bold cover line. Plan such things as whether your first page will start on the right-hand of the open booklet, which is the usual style, and use blanks freely, especially at the beginning or end, where they may be necessary or convenient to make a nicer arrangement of finished booklet pages.

Make a chart of your pages

The number of lines in the stack is the number of sheets you will need to reproduce. The arrangement of pages on the sheets above is for a booklet that folds downward, so the cover is at the top right, which opens to the booklet coverleaf (with a blank back), then lower-case Romans for the tables of contents and authorities (use more if your tables or similar introductory matters run longer). Page 1 falls on the right-hand side of the open booklet, and the Arabic text pages follow front-and-back to the appendix. The appendix coverleaf should also fall on the right-hand side, followed by a blank. Insert a blank at the end or in front of the booklet coverleaf if your page count is uneven.

Paper-clip the page proofs (LaserWriter output) together, assemble the stack, and turn through them to make sure it all looks right. Once you're sure it's just the way you want it, paste them up in order.



		□	
	blank		cover
blank	blank		blank
app 3			coverleaf
	app 2		blank
blank	app 1		i
Appendix			blank
	12		1
10	11		2
9			3
	8		5
	7		6

Make two-sided copies of the pasted proofs, referring to the chart and making a trial copy of each before running the full count. Double-check the margins and the matchup through the page. Front and back should cover the same areas of the sheet. If it's off line, move the second-side master around until the trial print matches up.

There's no easy way to collate this kind of work, unless you find a machine that can keep track of the fronts and backs while collating, so you will probably need to assemble the final stacks manually.

Photocopy the covers on an index or other suitable stock, or consider having the covers offset-printed as a separate job while you do the rest. Cut the stacks, with covers, down to the finished size. For a booklet of more than 20 or so pages, allow a little more than twice the finished size (a sixteenth of an inch) when pasting up and cutting to allow for a thick fold. A good sharp paper cutter (a.k.a finger remover) will do this, but the print shops have a machine that cuts stacks very cleanly. Do it yourself slowly and carefully or pay to have it lopped off in a trice.

Folding and stapling is easy on the machines at Copy-Rite, but the job is relatively simple to do manually. Staple through the cover toward the center sheet, so the ends of the staples are inside the booklet, and fold carefully along the staple line.

Now, put 40 of these into a big envelope addressed to the Supreme Court, put six dollars of postage on it, and wait a few months.

Zen and the Art of the Macintosh: Discoveries on the Path to Computer Enlightenment

Frank Patrick



This has been a summer of nostalgia for me. First I rediscover my Alma Mater through an alumni marching band (Rutgers University '73). Then a business trip takes me to central North Carolina, the scene of the summer of my youth (Greensboro, 1967). Last week there was a 20th anniversary salute to the Fillmore on HBO. And now I pick up a book that I expect to be about one of my current infatuations, the Macintosh, and find out that it is really about some of my old infatuations—Zen and the Tao; those holistic philosophies of my counter-cultural youth.

I'll admit it. Sometimes my tastes are just this side of weird. So when I saw this book at the Princeton MacFest and was caught by the title (so blatantly lifted from the classic *Zen and the Art of Motorcycle Maintenance* by Robert Pirsig, a tale of coming to grips with rhetoric and technology), I browsed through it and was totally blown away by the MacPaint illustrations. I knew I had to check it out more thoroughly.

Zen and the Art of the Macintosh, by writer/illustrator Michael Green, is one of, if not the first major publishing project "written, edited, designed, illustrated, typeset, laid out, and pasted-up entirely on a Macintosh computer." It is ostensibly the odyssey of Mr. Green in his introduction, fascination, and coming to grips with the graphic seduction of the Mac. Profusely illustrated with an eclectic, but heavily orient-oriented assortment of MacPaintings and modified digitizings, *Zen and the Art of the Macintosh* is highly reminiscent of the multimedia books of Marshal McLuhan and Buckminster Fuller from the 1960s (There's even a poster included, like my old Dylan album!!!). The media is the message/massage. The content is in the art as much as in the words. If not more so.

Green is one of those people who can use the phrase, "the rest of us," in full seriousness, without the slightest smirk. He can do a full functional specification on the concept of yin and yang, worthy of an IBM manual for the one side of your brain, and then fill the next two pages with a highly stylized version of the dots in the overlapping circles for the other side. Verbally and visually quoting the I Ching, the Tao, and various Zen masters, as well as numerous western artists and philosophers, like Descartes, Chagall, Leonardo, and Lewis Carroll, he draws the obvious parallel between the philosophical duality and the digital 1s and 0s that one might readily expect. But then he goes and draws another, unexpected, parallel between the early hacker ethic of free sharing of information and the medieval artists who shared ideas and images.

It's a frustrating book, alternately infuriating and fascinating.

Indeed, sometimes the turtle-necked, 1960s, Eastern philosophical sensibilities come through a bit too strong and get a bit syrupy. But just when that starts to happen, you turn the page and are fascinated by a either a little gem of verbal glitter (like the idea of "mental floss") or by a fireworks display of lasered graphics. Even the author admits this when, discussing his analysis of the left brain-right brain, analytical-holistic dichotomy, he writes that the



"Analytical You is scrutinizing these statements for flaws in accuracy right now. Holistic You has already lost interest in the discussion. It's— She's (!)— probably staring out the window, wondering when we're getting on to the next picture."

And what pictures! From highly detailed fat-bit edited manipulations to the simplest filled-region and sumi-style minimalism, the book is overflowing with images and ideas. And not only does he offer the graphics as support for his narrative, but in several cases takes the reader/viewer through his creative process, leading in and out of the trial-and-error path that the **Undo** and **Save as...** commands make so fertile for him.

The final chapter, "Nuts & Bolts & Bits", is a primer on the tools and techniques used in the artwork. Essentially MacPaint with Click-Art Effects, Pagemaker, and Easy3D, none of the tips are particularly new, but when told by an artist, with examples, one can really see their use. One which indicates the thought that Green puts into his tools, is the use of Pagemaker as a graphic tool, drawing beyond the boundaries of the page to attain huge, sweeping arcs otherwise only difficultly attainable. Not only the how-to but the why as well.

How-to and Why. Word and Pictures. 1960s and 1980s. Digital and Natural. MacLuhan and Macintosh. Hippie and Hacker. Yin-Yang and Binary Bits. Mr. Natural and Steve Jobs. Sumi and Fat-Bits. Switcher and Sutras. Mantras and Mice. Sitting Zazen and Midnight Hacking. Art and Science. Art of Science. Science of Art. Haiku and High-tech. Options and Option Keys. Tao and Technology. Infuriating and Fascinating. Angels and Architecture. Simon and Garfunkel.

Omigod . . . He's got me doing it!!!

Zen and the Art of the Macintosh is an intriguing book (I like the word "intriguing" — it can be totally meaningless in the right context— like a Zen koan!). It's not only for the Macintosh user, but also for anyone interested in creativity, philosophy, technology, and the arts. It would be sure to be appreciated by your average renaissance man or woman, or your average idealistic college kid (if there are any of those left around days), or your average ex-hippie who's thinking of getting a computer to go with his Cuisinart and condo.

But a warning— don't show it to your company's IBM/MIS department if you're still trying to convince them of the Macintosh as a tool.

They won't understand.

Zen and the Art of the Macintosh: Discoveries on the Path to Computer Enlightenment
Michael Green, Running Press



Tutorial: Silicon Press

Barbara J. Chan

Print mailing labels, cards, ID tags, invitations and disk labels in color?! Silicon Beach Software, creators of RealSound (*Airborne!* and *Enchanted Scepters*) and useful utilities such as those in *Accessory Pak 1*, released *Silicon Press* this spring for Macintosh color fans.

With *Silicon Press* you can lay out graphics and text to match any card or label up to 17 x 17 inches. By pasting in your artwork and selecting fonts for the text, you can design an attractive label or leaflet for both black and white or color printing on the ImageWriter I, II or LaserWriter.

When you want to print colored mailing labels for people whose names are stored in your database, you can import data from your favorite database file (*FileMaker*, *MS File*, *Word*, *OverVUE*, *Jazz*, and *Multiplan* to name a few).

I wanted to mail an announcement to some conservation corps members (Figure 1). Besides wanting to create the flyer in color, I wanted a colored mailing label with logo (3-1/2 x 15/16 inches) for 35 addressees. So I loaded *MacPaint* with Rulers (from *Accessory Pak 1*) and *Silicon Press* into *Switcher*.

Designing the Mailing Label

First design a logo 3/4 inch square in *MacPaint*, using the *MacPaint* icon. The page was to be printed in black, the paint streak and paint brush tip in red, and the paint brush and hand in blue. Using the lasso, separate the original drawing into three components: page, paint streak and brush tip, and brush and hand. **Save** them individually in a *MacPaint* document.

Next switch to *Silicon Press*. Set the mailing label design size at 3.5 x .94 inches and the page for **Portrait** (corresponds to **Tall** in *MacWrite* Page Setup) with the same measurements. If you are using an ImageWriter II, set **Preferences** in the Format menu for black and white screen and color ImageWriter II printer. That brings up an additional menu called Pen Color.

Now go back and **Copy** the page component from *MacPaint* and **Paste** it into the *Silicon Press* Design window at the lefthand side. Clicking on **Show Rectangles/Colors** under the Format menu brings up a dotted rectangle around each item in the design window. Now click your pointer in the middle of the page graphic, go up to the Pen Color menu and select black.

Repeat the process with the other two graphic components by copying and pasting them into the *Silicon Press* window. Before positioning them into the page component, click on each of the parts and select red and blue pen colors.

Now, using the pointer, move them into proper position. With the Show Rectangles/ Colors feature on, you can see exactly which components will be printed in what color. Figure 2 shows the mailing label and its components.

To type the address data format, select a font from the Font menu, click on the point size you desire, and choose a color from Pen Color. As with *Microsoft Word's* Merge, type in your field names, marking them with the data merge characters (option-backslash and shift-option-backslash):

BMUG Fall 1986



FALL GREETINGS

As an Environmental Conservationist, you have contributed to the health and welfare of our ecological environment.

We'd like to bring you up-to-date on how your contributions have helped expand the work of the Bay Area Conservation Network.



As part of our ongoing work, \$5,000 of the money you raised last year was devoted to three recycling centers in Oakland, Berkeley and Albany. The funds were used to upgrade existing equipment and to provide recycling containers for residential customers. Each of the cities where the recycling centers are located subsidizes the operation by providing people to staff the centers.

Because of our commitment to increasing the open space acreage for everyone to use, The Network donated \$10,000 to purchase the area now known as Purple Ridge Park in the East Bay Hills.

This year we will increase our fund-raising activities to include a Summer Auction, a Wildlife Slide and Sound Show by Bill Hickock and a Mud Mime Troup play about the environment.

Won't you join us in these and other activities? For more information, contact Bay Area Conservation Network at 415-ECO-LOGY.

Artwork courtesy of Real by Electric Cottage Industries

Figure 1

«name»
«address»
«city», «state» «zip»

MAILING LABEL COMPONENTS

«name»
«address»
«city», «state» «zip»

Page: Black
Paint streak and brush tip: Red
Hand with brush handle: Blue
Address: Green

Chickadee eye: Red
Beak: Orange
Head: Black
Breast: Orange
Upper Wing: Blue
Lower Wing: Black
Underbreast: Yellow
Pine Bow: Green

CHICKADEE COMPONENTS



Figure 2

Figure 3



«name»
«address»
«city», «state» «zip»

Now it's time to test your design using **Test Print** from the File menu. **Test Print** prints your design with a dotted rectangle the size you have indicated in the **Set Design Size** feature, and it helps you position your design on the mailing label. My design was created for roll labels 3-1/2 x 15/16 inches. If your design is not correctly aligned, reposition your mailing labels and try it again until it is perfect. Then **Save** the design under the name "Mailing."

Next set up the data file for the print merge. Quit *MacPaint* and load your database application in *Switcher*.

My records are stored in *FileMaker* and include more than 13 fields. I only wanted to print five fields. I sorted through my database and selected the 35 names to whom I wanted to mail my flyer. With *FileMaker* it's easy to output the data to an ASCII textfile with the name, address, city, state and zip fields selected. The data was saved under a new name, SP Merge.

Switch back to *Silicon Press*, open "Mailing", and click **Open** from the File Menu. Next **Open SP Merge**. The contents flow into the Data Window. To format it for print merge, clean up the tabs and separate each field with a comma and space. Once that is done, click on **Print** from the File Menu and your labels will be printed one by one, with the proper data recorded for each addressee. Note that there is no separate Print Merge command in the file Menu. You must import your data into the Data Window for the merge feature to work.

Creating the Colored Flyer

Set your flyer design at 8-1/2 x 11-1/2 inches under Page Setup. Under the Format menu **Set Design Size** at 8 x 11 inches.

I chose a chickadee from Electronic Cottage Industries' *Real Art*. As with the mailing logo, you separate the graphic into its color components (Figure 3) and position them in *Silicon Press*.

Remember how we can't wrap words around graphics in *MacWrite* or *MS Word*? *Silicon Press* solves that problem by allowing you to lay out graphics in different color components, and to type text anywhere on the page in the color of your choice. By typing your text in blocks and placing the blocks wherever you like, you can get the word wraparound effect (see Figure 1).

If you type each block of text separately and choose your color, it will be easy to lay out your page design. Voilà, a completed color flyer!

Comments

The *Silicon Press* manual is logically and thoughtfully organized with ample illustrations and begins with a brief tutorial called A Quick Look at *Silicon Press*. The bulk of the documentation focuses on using the program. It concludes with appendices to show you how to use *Silicon Press* with *Switcher* and with keyboard commands.

I would like to see a tool box similar to that found in *MacDraw*, *MacDraft*, *MacPaint* or *PageMaker*. Granted, *Silicon Press* is not meant to be a full-fledged page layout program, but it could use line, box and circle drawing tools, and several shaded fill patterns. Lines and boxes can be effective design enhancements.

BMUG
Fall 1986



At \$79.95 *Silicon Press* is a bargain for anyone who wants to create colorful leaflets and flyers. For more information, contact Silicon Beach Software, P.O. Box 261430, San Diego, CA 92126 (619-695-6956).

Other Products Mentioned:

FileMaker
Forethought MacWare
1973 Landings Dr.
Mountain View, CA 94043
800-622-9273

Real Art
Electric Cottage Industries
P. O. Box 217
Spooner, WI 54801-0217
715-635-3516

Barbara J. Chan is a freelance writer, trainer and desktop publishing consultant. She can be reached via GENie: BJCHAN, The Well: bjc, or 415-644-2794.



circle/Chan

Msg. #4177 in **Public Messages** Posted on 10/31/86 at 09:09:16

To: ALL From: MARK CHANDLER

Subject: MacDraft v1.2a

Here's a few tips to users of MacDraft v1.2a:

- 1) If you hold down the TAB key while drawing a line, you will get a dimension line (rather than having to choose it from the LINE menu.
 - 2) If you are having problems selecting fill patterns, power down ALL hardware, remove your battery for about 30 seconds, and then boot back up (not sure why it works, but it does). Also, if you are using old control panel, the blink rate (upper right setting) should NOT be turned off.
 - 3) Older 1.1 documents can't be opened from the desktop, you must open them from within the program itself (from the file menu). The creator type was changed (from MACD to MDFT).
- Hope these help,
Mark Chandler



SUPERPAINT: Its Name is True

Dan Wood

I first saw SuperPaint in its earliest form back in January 1986 when the programmer, Bill Snider was showing it off for Silicon Beach Software. *SuperPaint* is an extremely well-thought-out combination of FullPaint and MacDraw, with improvements on both.

SuperPaint gives you a pattern palette and a tool palette on the screen like *FullPaint*, and allows you to open multiple documents. The key to the program, however, is that it actually consists of two *layers*: a bitmapped, painting layer (like what *FullPaint* or *MacPaint* produce) and an object-oriented layer (like what *MacDraw* or *MacDraft* produce). The layers coexist in harmony, sharing almost all the same tools.

A Background

Although bitmapped objects and draw objects that appear on the screen may look the same, they have quite different characteristics. A painted circle, for example, is a mere collection of dots (which can be changed with the spray can, the eraser, the pencil, etc.) that will print with the "jaggies," while a drawn circle is stored as a circle with a particular fill and pen pattern that can be resized or reshaped at any time and will print very smoothly on a laser printer. Similarly, a painted block of text prints with the jaggies but can be modified graphically, while a drawn block of text will print with beautiful laser fonts and can be edited at any time.

The fact that the different characteristics of bitmap (painted) versus object (drawn) objects are needed for different reasons makes the availability of both kinds extremely useful.

As an example, most text in a document should be created on the drawing layer, so that they will be printed in laser fonts. Text with special effects of some sort are better off being painted, so that they can be modified. A polygon representing some geometric form should be drawn; a polygon which is the basis for a picture should be painted.

While the ability to combine of painted and drawn graphics was available to some extent in *MacDraw* (using *MacPaint* and *Switcher*), the process was time-consuming; the resulting graphics were never quite right. *SuperPaint* is a program which was carefully written to ensure that everything comes out just right. The scaling of drawn and painted objects align *exactly*. A painting with a drawn border will be aligned when printed just as it is aligned on the screen. Accomplishing this was impossible in *MacDraw*!

The Macintosh screen displays 72 dots per inch. Since the LaserWriter displays 300 dots per inch, *SuperPaint* makes one dot on the Mac screen print as 1/75th of an inch (4X4 LaserWriter pixels). Additionally, with a wonderful feature called LaserBits, a painted object will be printed at 300 dots per inch (more on this later).

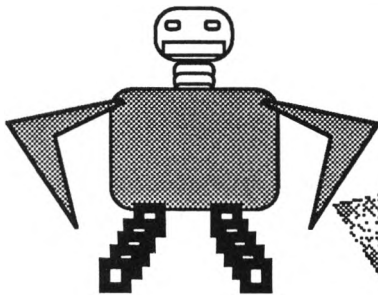
What the Program Will Do

SuperPaint has almost all of the features in both *FullPaint* and *MacDraw*. The Basic tools—ovals, rectangles, lines, polygons, arcs, freehand shapes, fill patterns, line patterns, and so on, are all there. Other basic tools which only apply to painting (since they deal with bitmaps) include selection boxes, lassos, fatbits, erasers, spray cans, and paint buckets. Tools and concepts which apply only to objects include resizing and reshaping, grouping, bringing to front and sending to back, and true gray shades.

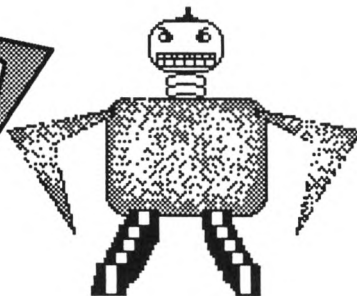


Here are some other amazing features in *SuperPaint*, which I will describe in sentence fragments: Filling characters and character backgrounds with gray patterns. Scaling drawn objects by an entered percentage. Selecting graphics for cut & paste from both layers at once. Full screen graphics (like *FullPaint*) with the ability to hide the tool palettes and the menu bars. Auto Scrolling for drawing or selecting of large objects. Drawing shapes (rectangles, circles, etc.) from the center out. Double-clicking the lasso for lassoing of the whole document. Drawing a selection box around bitmaps that will turn into a lasso or shrink into a box around the edges. Constraining lines not only to 45° and 90° but also to 30° and 60°. 32 gray shades. Dragging complex objects as outlines to speed things up. Adding vertices to polygons. Being able to open and save as MacPaint, PICT, and startup screen documents. Arbitrary font sizes. Color printing on ImageWriter II. Grabber (hand to move about the document) available all the time. 2X, 4X, and 8X magnification. Editable brushes and patterns. Multiple pattern palettes. Lines that can be of varying horizontal and vertical thicknesses. Free rotation, stretching, slanting, skewing, and perspective (like *ClickArt Effects*) on paintings. "Nudging" a selection to move one pixel at a time. Rulers.

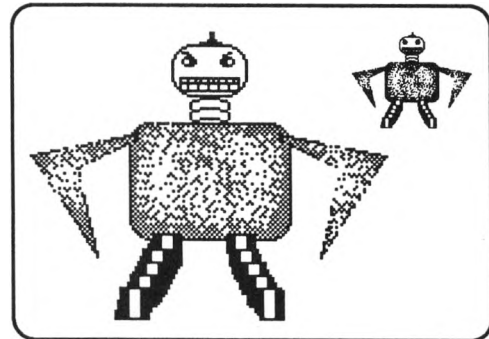
Since *SuperPaint* works on two planes, you can transfer an object from one plane to another. For example, if you have drawn (using objects) a robot, using rectangles, and you want to then add various shadings, you can select the objects and copy them onto the painting layer. The image (as you see on your screen) is now in bitmap form, which can be altered with the spray can or the eraser.



The robot was first
created as a
collection of objects



then moved to paint
layer and edited



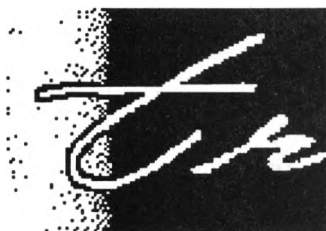
then moved back to draw layer,
also reduced 33%, on a gray box.

Transferring in the opposite direction is equally useful. If you want to put the picture of the robot on a gray background, you would select the picture of the robot with the marquee and transfer it into an object. The robot will now be seen as a drawn object (with handles) which can be placed upon a gray rectangle. An object which is actually a bitmap image is fairly static; it can't be modified with the painting tools. The real beauty of moving painted object onto the graphics plane is that they can be shrunk or expanded without losing any resolution when printed on the laser printer. No longer are you forced to create bitmapped graphics that reek of Mac— just draw something bigger than you want it to appear, shrink it, and print it on the LaserWriter.

SuperPaint takes this concept of high-resolution bitmapped images one step further. With LaserBits, a separate bitmapped picture (stored in a different *SuperPaint* file) is linked into a *SuperPaint* file to be printed at 300 dots per inch, the resolution of the LaserWriter. What you see on the screen is only an approximation of a very sharp image. The image is actually just a bitmap picture printed at 25% actual size. Thus a *SuperPaint* document which is 8 inches by 10

BMUG FALL 1986

inches will be 2 inches by 2.5 inches on your document. Since the LaserBits image is stored in a separate file, the document can contain many of these images without hogging the Mac's memory.



part of original



LaserBits image on the
screen



LaserBits Image when
printed

SuperPaint is, without a doubt, the best graphics program available for the Macintosh. If you create graphics, you will make a lot of use of this program. This is, of course, only the first version of the program, and I can think of several more features I'd like to see in an upgrade, such as arbitrary rotation of objects, tracing outlines, reshaping rounded rectangles, masking bit images (like ComicWorks), entering the dimensions of an object, and producing real, editable PostScript code (which would allow rotation of text, thinner line widths, and special effects), like *Cricket Draw*. But then, these features would make it the *ultimate* graphics program. For now, I'm happy with just the penultimate.

Note: As of this writing, *SuperPaint* has just been released; I am reviewing version 0.95, available for customers who can't wait for version 1.0. It lacks five features scheduled for 1.0: color printing, rulers, bitmap effects like rotation and stretching, arbitrary text size entry, and grids. By the time you read this, version 1.0 or greater may already be out. Charlie Jackson, the president of Silicon Beach Software, says that future versions should have more features. The company is very accommodating in handling the requests of the public, so if you have any comments about what you'd like to see in the program, drop them a line.

SUPERPAINT

Silicon Beach Software, P.O. Box 261430, San Diego, CA 92126. (619) 695-6956.

List price \$99.95.



A Preview of Cricket Draw

Dan Wood

Cricket Draw, by Cricket Software, the makers of *Cricket Graph*, is an object-oriented graphics program like *MacDraw* with a sophisticated twist. As of this writing it hasn't been released, but I saw it in demonstrated at the MacWorld Expo in Boston and the Seybold Desktop Publishing Conference in San Francisco.

The special thing about *Cricket Draw* is that instead of sending its output to the LaserWriter by the standard method (using the LaserWriter driver to encode it into a pseudo-PostScript gibberish that the LaserWriter can barely understand and most humans cannot), the graphics are converted into a more-or-less understandable, *modifyable* PostScript file which is then sent to the LaserWriter.

So you don't program in PostScript? You're not alone. The main advantage is that the program can generate much more accurate graphics that look much better. If you've used *MacDraw*, you've probably wished for thinner or thicker lines or more shades of gray. *Cricket Draw* supports that. PostScript also allows for dashed lines of various patterns, which I believe is also supported directly from the program. And, of course, any item (including text, you *MacDraft* fans) can be rotated to any value, and slanted, such as turning a rectangle into a parallelogram. What was impossible in *MacDraw* is trivial for this program since it sends direct PostScript.

The tools available on the program are much like *MacDraw*— Polygons, text, circles, arcs, and so on. There was a strange little printer icon at the bottom that has something to do with bitmapped graphics.

Cricket Draw also has some "special effects," which consist of starbursts (a circle made up of lots of lines radiated from the center), horizon lines, and "fountains" (filling an object with gray that starts light at one end and gradually turns darker towards the other end). These are very nice, although I dread to see the next generation of documents that *reek* of Macintosh by having fountains and starbursts all over them! Hopefully Cricket will be coming out with a library of other PostScript goodies for some variety.

Once the document is created, a window can be displayed that contains the actual PostScript for that document. Each object on the screen corresponds to a few lines of straightforward PostScript code. If you had created a triangle filled with a particular shade of gray and a particular line thickness, you probably wouldn't have much trouble finding that on the PostScript listing and seeing what corresponds to what, then adding a few embellishments if desired. The coordinates of the objects look somewhat complicated (such as moving to point 312, 412 and drawing a line to 330, 446...you get the picture) since those numbers aren't too revealing about the location on the page. Hopefully the final program will have the option of giving coordinates in inches or centimeters so that the position on the page will be more intuitive.

I am very excited about this program! As a PostScript programmer, I have had to suffer with the tedium of laying out a page with a lot of guesswork. This will bring the "what you see is what you get" aspect of the Mac (or, as they say, "what you get is better than what you see") to PostScript programming.



The person at Cricket Software with whom I spoke told me that the graphics created with *Cricket Draw* (including the unique effects) could be cut out and pasted into other applications, retaining their PostScript compatibility. I certainly hope that's true! I doubt, however, that any changes that you make to the PostScript text get reflected in graphic document. The sample page which they handed out contained an airplane that was created in *MacDraw*, so you won't have to start from scratch.

If you do graphics on the Mac, check out this program when it's available. It's on *my* Christmas wish list.

CRICKET DRAW (not yet released as of this writing, due Fall/Winter 1986)

Cricket Software

3508 Market Street, Suite 206, Philadelphia, PA 19104. (215) 387-7955, (800) 345-8112

TidBytes by Ken Hule

History of VaporWare

The term vaporware may be a newly coined term but the concept is centuries old. In 1777, at the end of a self-published pamphlet, George Palmer promised a future "Chemical Treatise on the Art of Dying" and a transcript of a discussion with Samuel Johnson "concerning the production of black cloth." They were mentioned as forthcoming by several other authors for the next twenty years but neither of the two products ever came into being.

A Speedy & Simple Serial Signal Stethoscope

An undocumented feature of the ImageWriter I is the ability to convert all ASCII received to hexadecimal format, 24 hex numbers to a line. This ability is useful when you need to test a terminal program, debug a print routine, or you just want to find out exactly which control codes are being sent out through serial port.

To get into this mode, hold the LINE FEED button down when you turn the printer on. When there are less than a line's worth of characters in the buffer, pressing SELECT will print those characters received. Get out of this mode by cycling the power off/on again. This function is a ROM holdover from the time when the ImageWriter was known as the ProWriter (built by C. Itoh). It was not documented then either.

Custom Airborne!

The game Airborne! has a file called StartupSong that can be replaced with any MacRecorder file shorter than 160K. Make the MacRecorder file from a speeded-up recording since Airborne! will play back at a slower speed. Use SetFile or FEdit to change the Type to WAVE and Creator to WVED, then copy the file to the game disk. There is nothing like a rousing chorus of Louie Louie before doing battle with the infinite horde.



ComicWorks & GraphicWorks

Two Programs, One Review, Two Opinions.

Dan Wood

The program(s)

ComicWorks, from MacroMind and Mike Saenz, the creator of the *Shatter* comic books, is a program that allows Macintosh Users to make comic books, cards, ads, posters, newsletters, and comic strips. Essentially, it is a combination of *MacPaint* and *MacDraw*. *GraphicWorks*, on the other hand, is essentially the same program with a different title. MindScape, the company distributing *ComicWorks* and *GraphicWorks*, has released the same program under two different titles and packages so that non-comic-artists will not be deterred by an unserious title.

The software packages are not very different. The main differences are that the templates provided with the program are essentially space-hero pictures in one, page templates in the other (can you guess which one is which?); and that the manual has been rewritten. The software, except for a few minor differences, is functionally the same. Thus, I will be referring to the *ComicWorks* program, although I am actually referring to both. Confused? Good.

What it does

ComicWorks works on the surface much like *MacPaint*, in that you are given a number of painting tools to draw a picture. Rather than working in a page, you work on an easel. That is, you only draw one particular object within the confines of an easel. Several easels fit within a panel, and several panels will fit within page. Imagining the page of a comic book, the terms are fairly clear.

The painting aspect of *ComicWorks* is very easy, since it is almost the same as *MacPaint*. One feature in the program that I absolutely love is the airbrush. It is similar to *MacPaint*'s spray can, but the airbrush paints with an intensity depending on how long the brush is held down in a given spot. It makes the old spray can *obsolete*. The other new paint feature is ThinBits™ (was FatBits ever trademarked?), which allows you to draw while in the Show Page display. With this feature, you can draw something big with the right proportions, then clean it up in normal size or FatBits. Anybody who has tried to draw a full-page figure in *MacPaint* will love this feature.

The way that easels work within a panel is *ComicWork*'s most interesting feature. Easels overlap, so that objects can be drawn individually and placed in the correct place on a panel. For instance, a panel might contain a mostly-black field of stars on the bottom, with a planet, spaceship, and rocket-blast tail on top. Overlapping bitmap images is no trivial matter, since overlapping black and white dots should produce the correct effect. Given a picture of a person standing in front of his airplane, you shouldn't be able to see through the person! *ComicWorks* deals with this by having objects be masked— that is, an outline of the topmost figure is essentially erased before placing the outlined figure on the page. All the standard methods of bitmap transferring on the Macintosh are also available: COPY, OR, XOR, BIC, etc., allowing for some special effects.

The other main feature of *ComicWorks* is the "voice balloon" universal to all comics. *ComicWorks* provides a wide variety of balloons (that can be reshaped and stretched at any time) containing any mix of fonts and styles (Hooray!). A voice balloon exists on the same plane as an easel; it can be moved around on a panel.



The templates

Two disks of templates come with the *ComicWorks* program. Essentially these are only useful for inspiration, unless you want to make a space opera. There are several views of a couple of spaceships, several pictures of a serious, muscular astronaut, probably named Dirk or Lance, a shapely blonde space woman, grotesque BEM's (bug-eyed monsters), Dinosaurs, planets, alien landscapes, explosions, and star fields. The artwork is probably the best bitmapped Macintosh graphics around! Additionally, a sample comic book (featuring the same ships, aliens, and humans), figures in motion, and sample greeting cards and newsletters can be found on the template disks. *GraphicWorks* comes with one template disk of typical desktop publishing templates.

How does it handle?

ComicWorks is often very clumsy and frustrating for a beginner. The concept of panels and easels takes a bit of getting used to; even with that, it's a major hassle. Easels have a tendency to disappear under other easels when you want to use them, or to persist on staying in front when you want to get rid of them. Response can be a bit slow at times. And—horror of horrors— I got a system bomb when reshaping a panel, and another one while printing.

The bad news

ComicWorks' main problem is its terrible printing on the LaserWriter. In reading the manual, I was discouraged from using the detailed print mode, in which laser fonts (like Times, Helvetica, etc.) and balloons are printed on the laser printer as characters and smooth lines rather than as jagged dots. Ok, so they don't want me to do it. Why? The output is terrible. The bitmaps in the panels don't line up properly in relation to the panels. Some LaserWriter fonts were improperly substituted, and the test got repeated not unlike extended versions of pop songs. Panels run off the page. Isn't the quality supposed to be *better* on a laser printer? When printed in the regular mode, you the quality isn't any better than that of an ImageWriter, since the laser fonts aren't substituted and the angled lines in balloons look jagged. At least Silicon Beach Software, in creating SuperPaint, was nice enough to circumvent such problems when dealing with bitmapped graphics.

Should you get *GraphicWorks*?

In my opinion, *GraphicWorks* is *not* the program to buy for desktop publishing. If you happen to have *ComicWorks* and are using an ImageWriter, fine.

Ok, so should you get *ComicWorks*?

If you want a good environment for making comics or storyboards, get *ComicWorks*. In spite of its faults, the way in which bitmapped objects can be placed upon each other makes the program very nice. Realize that you will have a fight ahead of you if you wish to use the LaserWriter, and that the output will "look Macintosh." For making comics, I found this program fun and addictive.

COMICWORKS or GRAPHICWORKS

MindScape, Inc., 3444 Dundee Rd., Northbrook, IL 60062. (800) 221-9884.

List price: \$79.95

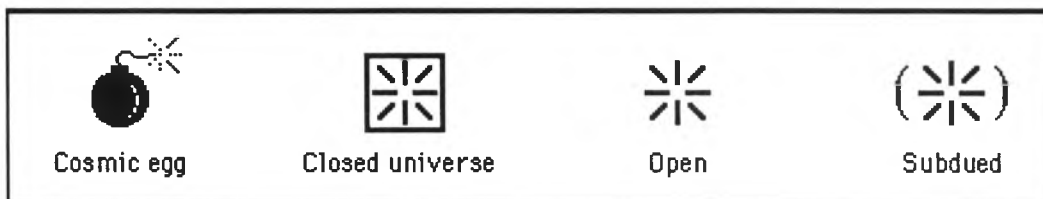


Towards a Strategy for Macintosh Explosions

Glenn Harris

Yes, destruction-graphics really *can* add focus to that otherwise text-heavy quarterly report. But realize that, through overuse, this once-innovative technique now requires a particular *savoir-faire* in order to lob your own fulminations above the current sloppy parade of pyrotechnics.

What follows is only a brief summary of techniques and helpful tips. If you are already an old hand, you may still find some of the subtler points helpful in your drawings. If you are new, welcome to the rewarding world of expressive diagrammatic explosions.

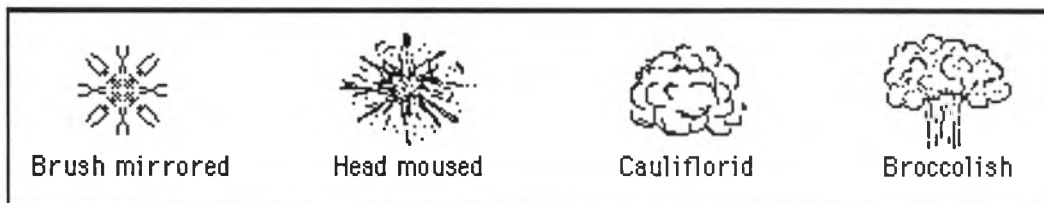


The Basics

- Misplaced and badly timed explosions can destroy vital data and implies an unprofessional attitude.
- Mix explosion graphics with other visuals. Overdependence can turn this serious business tool into just another gaudy effect.
- Use MacPaint or FullPaint, not MacDraft or MacDraw. Good explosions require a flexuous touch - not a ruler.
- Keep away from digitized pictures or clip art blowups. But digitized clip art is OK.
- Take care **never** to compile explosions. If needed, use them only in interpreted code.
- Don't live your pixels. Keep the violence on the screen.
- Balance this with a healthy amount of raw aggression and artistic obsession.

Time and Motion

- Explosions should not look like snapshots, but must embody the illusion of motion. To this end, a number of cheap props are available: lightning, sound waves, smoke, particulate matter, shadows, stretch marks, flying technicians, and punctuation.
- Use brush mirrors sparingly - it's good for basic structure, but its symmetry lacks drama.
- Laserwriter users: turn that smoothing off!



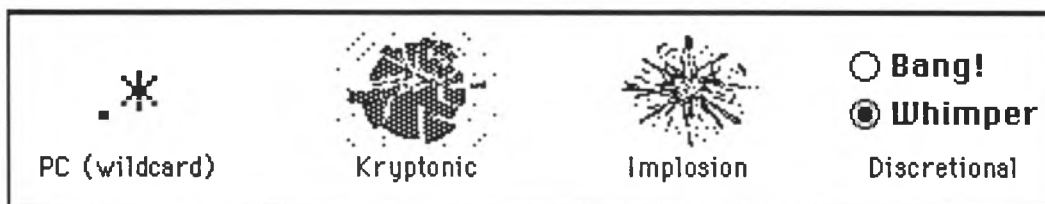
Subject matter

- Avoid bad taste whenever possible. Don't blow up household pets, children under 12, automobiles, Sandinistas, or shareware.

- Trim overlage subjects to provide for efficient use of the full blast intensity.
- Subjects should mean something to the surrounding document: e.g., ice cream with pie charts, livestock with stock reports, and squid with BMUG newsletters.
- Centralize the subject - keep them away from the border cells in your spreadsheet and from the footnotes.

Verisimilitude

- Always draw the object intact first. Use the lasso for displacing parts (insuring conservation of mass).
- Establish the explosion definitively in time and space. Include clocks, calendars, and landmarks in the background.
- Motivate unconvincing blasts with one or more of these foreground items: lightning, a plunger, commandos, restless crowd with torches, unreasonable political demands, a film crew.
- Gravity should pull ejecta **towards** the nearest planets. Northern hemisphere discharge rotates **clockwise**. Human-generated explosions can occur only **after 1200 BC**. Finally, remember any vacuum effects if in space.



Nucleus

- You can always tell the work of a pro by the look of that explosion nucleus. Spend at least 40% of your total drawing time on it.
- Never have more than one nucleus per explosion. Double-check this before any important presentations.
- The dark/light nucleus question: the experts will never stop arguing about this one, but a light center does tend to suggest formal occasions.

Variations

- To add variety, try implosions. They do require more practice than their more popular cousin, so try them out first on internal reports. Just follow the above tips in reverse.
- There are rare instances when you've got to "say it nuclear." But those techniques are beyond the scope of this article.



High Speed Animation on the Mac: A Review of the Beck-Tech MacMovies Program

Philip J. Suh

There are many graphic programs out on the market for the Macintosh, but none quite as unique or as powerful as MacMovies by Beck-Tech. MacMovies creates high speed animation sequences using ordinary MacPaint files. Unlike VideoWorks which animates with sprites, or individual screen elements, MacMovies animates full Paint screens all at once. Using MacMovies is analogous to creating cartoons the traditional hand drawn way. That is, it repeats the following sequence: draw each frame individually, take a picture, draw another frame with the elements in the frame moved a little, etc. MacMovies takes pictures of each frame and stores it in memory, thereby enabling rapid playback of the sequence.

In VideoWorks, having a large number of elements on the screen at once will noticeably slow the animation down. In MacMovies, because screenfills of data are manipulated, rather than individual elements, there is a barely perceptible amount of slowdown when complex screens are animated. The top speed of animation achieved by MacMovies is around 30 frames per second which is roughly half the speed that a T.V. images is refreshed (60hz which translates 60 frames per second). As you can imagine, at top speed, many frames are required to create an animation sequence lasting even a short amount of time.

The size of the finished movie is only limited by the size of memory available. Because it is necessary to store the whole movie in memory, it is mandatory to have at least 512K of memory, and more if possible! When creating the movies, the Film Compiler compresses the individual screens by only recording the changes that occur between frames. This way, whole screens (22K), do not have to be saved, merely those parts of the frames that differ from the previous ones.

With the high speed and large number of frames involved, detailed and very smooth animation can be achieved. It is not unlikely to have a movie composed of as many as 90 or 100 different MacPaint images. In addition to lots of memory, lots of free disk space is also a must. With an average Paint document size of 10K, a complete movie may require 1 Meg, of free disk space for just the raw movie data...this may be compiled down to as little as 300K to 400K of compiled data.

You may be wondering how is it possible to create 100 different MacPaint images. If done manually, truly, it would be an impossible feat, but with the Macintosh, many things are possible! Some of the drawing programs, particularly the three-dimensional programs will allow you to move your drawings just a little at a time. This is done by selecting slightly different views from within the program. For example, with Easy 3-D (the best 3-D program out on the market, in my opinion) you can rotate a drawn three-dimensional figure around 3 different axes in 1 degree increments. The way to create a movie using this technique would be to draw a figure, rotate it, and then take a screen dump, and repeat this process until enough frames have been created.

What are the advantages in using MacMovies? Well, with MacMovies, it is possible to create very professional looking animation sequences which are very smooth and fast. Also, if many objects are on the screen at once, speed is not sacrificed. In fact, since screens of data are animated, with the help of a digitizer, digitized movies can be created using actual pictures of people or things in action.

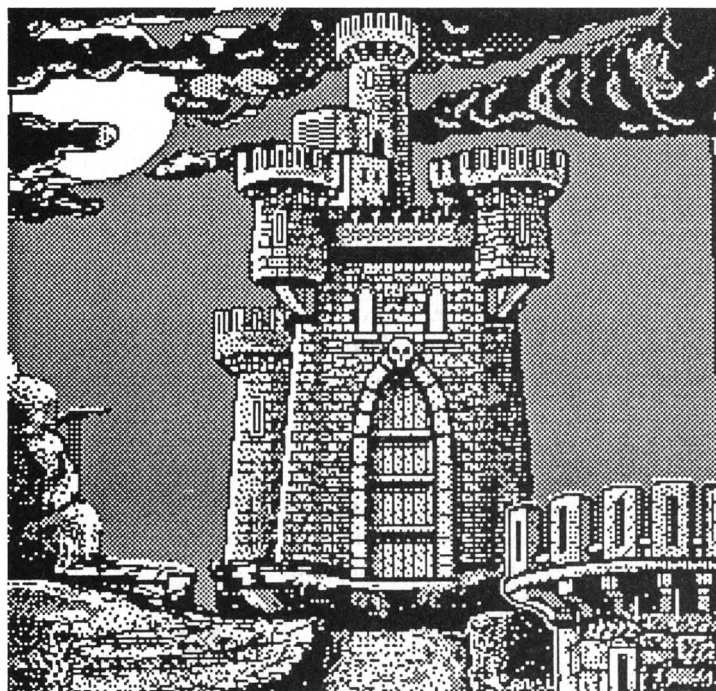


BMUG FALL 1986

Disadvantages? In order to get the most out of MacMovies, I think that you should have at least 1 megabyte of memory if possible. Also, you should have some way to draw out sequential scenes as MacPaint files, such as a 3-D program. In fact, you can speed up VideoWorks animation by taking screendumps of actual frames of VideoWorks, and then recompiling them in MacMovies. If none of these are available, I suppose you can spend hours and draw each of 30 to 100 screens out manually in MacPaint (Actually FullPaint is much better to use if you have it).

MacMovies has not been advertised or promoted heavily. I believe that awhile back they had an ad in MacWorld, but then again, how can you show off an animation program in a magazine ad?!! MacMovies retails for \$99.95, but you can find it cheaper through mail order houses. If you want to create very high quality animation sequences for displays or presentations, or just for fun, and VideoWorks just isn't cutting it, then I would highly recommend MacMovies.

For more information, contact Beck-Tech at 41 Tunnel Rd., Berkeley, CA 94705, 415-548-4054.



Dark Castle screen



How 3D graphics is done on the Mac.

Greg Simons

The aim of this article is to give the reader a feel for what is going on **mathematically** and **graphically** inside the various 3D engineering, scientific, and entertainment programs currently receiving a lot of attention.

This article is an *overview* of 3 dimensional computer graphics, with several specific references to the Macintosh and the 68000 microprocessor. Some attention is given to efficient algorithms.

We will walk through the high-level processes required to draw realistic rotated, scaled, and clipped line segments given the endpoints in a global coordinate system.

The viewpoint will be defined by two things: our position in the world (given in x,y,z), and our view angle upon the world (roll, pitch, and yaw).

The basic steps involved follow:

Translation Deciding what point to rotate around (modifying the origin).

Transformation (rotation) Computing the new (rotated) xyz point given the old point and the degrees of xyz rotation.

Clipping to the view volume Deciding which lines will not be visible, which ones will be partially visible (and removing the unneeded part), and which ones will be wholly visible.

Scaling (perspective) Making more distant objects appear smaller.

Projection Getting the 3-D coordinates onto the 2-D screen.

Line drawing Drawing lines between the now translated, transformed, clipped, scaled, and projected points (on a bit-mapped screen).

Think of the above as a filter with 6 sieves. The matter being passed through the filters are line endpoints. These points may be described in a 3-D *world* cartesian coordinate system. *World* coordinates means that all points in the world (in our simulated world) are described in the same coordinate system.

Inputs to the 3-D Filter: Your eye's location, given in three components: **x** , **y** , and **z** ; and your eye's angle in the world given in three components: **x -angle** (angle about the x axis), **y -angle** (angle about the y axis), and **z -angle** (angle about the z -axis).

These will be the parameters that are used in setting up the filter. After the filter is set up, we can pass the world coordinate points through and get a realistic 3D representation of that on the screen.

Translation: This step entails translating the (world coordinate) point passing through the filter to be in our own private coordinate system, where the **x** , **y** , and **z** of our eye upon the world is at the center (yes, programmers get to play God in this world!).

Simply put, this means subtracting our eye's (xyz) from the input (xyz). This sets us up for the next step: *transformation* (rotation).

Transformation: This step rotates the point by the X,Y, and Z angles about the origin. Since we just made our eye the origin, it will rotate about our eye.

(Real World Example: This is the case in flight simulators, where you perceive the world to be moving and pitching and rolling *about you*; CAD programs translate the origin to the middle of the piece to be rotated, so it is *magically rotating about its center* on a simulated drafting table.)

Mathematically, this is the most complex, and computationally the most time consuming. The idea is that a 3-D transformation can be thought of as a composition of *three* 2-dimensional transformations. First, we rotate the point about the x axis by the x-angle amount, then the y, then the z.

The x-angle formulas follow:

```
X' <- X      {no change to x, because rotation was along x axis}
Y' <- Y * cos(x-angle) - Z * sin(x-angle)
Z' <- Y * sin(x-angle) + Z * cos(x-angle)
```

The y-angle formulas are:

```
X' <- X * cos(x-angle) - Z * sin(x-angle)
Y' <- Y
Z' <- X * sin(x-angle) + Z * cos(x-angle)
```

The z-angle formulas are:

```
X' <- X * cos(x-angle) - Y * sin(x-angle)
Y' <- Y * cos(x-angle) - X * sin(x-angle)
Z' <- Z
```

How it's really done: To multiply every point by the x, y and z formulae would take 12 multiplications (integer multiplication on most modern CPUs takes about 10-20 times as long as an add--don't even think about floating-point without a math co-processor).

We can reduce the number of multiplications for each point to 9 (a 25% improvement) by creating a rotation matrix. This is done by making each of the above sets of equations into a 3x3 matrix and multiplying the three together into one master transformation matrix.

Creating the rotation matrix requires alot of multiplication, but it's worth it if you have more than a few points. Get a book on linear algebra or computer graphics for the details (too much to print here). Don't fear, it's a pretty mechanical process.

Performance notes: Unless super-high precision is required, a fixed-point representation for the trig values is typically used. Coding the matrix multiplication in assembler is best. Also, a table of those values should be used (no Taylor series computations on the fly!!).

At this point in the sieve, the points have been transformed into a new coordinate system, with one axis perfectly horizontal with respect to your eye, another perfectly vertical, and the third perfectly orthogonal to the others--pointing from the center of your eye out to (what will become) the vanishing point. At this point, we know alot about how the line will look: whether it is behind us, far above us, way out to the left, and so on.

Clipping: In this step you eliminate all lines and parts of lines that won't be within the screen boundaries. Each line falls into one of the following 3 categories:



- 1) wholly on the screen (both points on screen--just keep it),
- 2) wholly off the screen (**both** points above, below, left or right), or
- 3) partially on the screen (the most complex case).

Clipping is usually combined with scaling and projection, because clipping implies knowledge about how the points will map onto screen coordinates.

Determining what part of a line is actually in the view when both endpoints lie off the screen (say it might be crossing a corner of the screen) is not easy to do fast. In my opinion, this is the most difficult and least agreed upon area of high-speed 3-D microcomputer graphics, and where elegance is sometimes left in the classroom.

Scaling (perspective): This step has the most dramatic effect for the work that it requires. In translating and transforming the points, we re-defined them in our eye's coordinate system. One of the axes is perfectly horizontal relative to our eye (call it x), another is vertical (y). The third axis (z) is running right through the center of our eye (ouch!!).

To get perspective scaling, we divide each point's x and y by its z. Points that are further away in z converge upon the origin (in x and y). A line 10 units long at 10 feet will be 20 units long at 5 feet and 100 units long at 1 foot. Try proving that this the case in real life.

Some interesting wide-angle and telephoto effects can be simulated by adding to z and/or multiplying or dividing z.

Projection: This is the final process of getting the 3-D points into 2-D screen coordinates. Remember that we just divided X and Y by Z to simulate perspective.

To project to the screen, we simply drop Z (the axis that sticks straight out from the center of our eye) and use the (X,Y) pair to address the screen.

Final Tuning Before Line Drawing: If you find that your y is oriented the opposite way from your screen's y, you might modify the signs of a couple of elements in the transformation matrix (the same goes for x).

Since screens are typically addressed in terms of pixels, and your coordinates are probably in some unit of measure, you will have to scale them up. You could multiply by a ratio of *pixels* to that *unit of measure* (in our case of feet to the Mac screen, 256 seems reasonable - powers of 2 let us perform left shifts in place of a multiply--saving alot of time). You should perform this *before scaling*, else you will find that points grow and shrink in gradations of 256 pixels (not good).

Once you have processed 2 line endpoints, you are ready to draw a line.

Line Drawing: By the time you get here you've done a heck of a lot of computing: for the two endpoints, you've done at least 18 multiplications, unnumerable adds, subtracts, shifts, table lookups...and alot more heavy stuff in the case of clipping.

If speed is not an issue, then use QuickDraw. If it is (and it always is), then read on. QuickDraw is powerful because of its generality, not because of its speed. Many 3-D programmers would gladly trade away clip-rects, pen modes, patterns, and maybe even pen width in favor of speed.



*Line drawing takes **more time** than all previous steps combined!!!* In a bit-mapped environment, an arbitrary line that crosses the standard 512 x 342 pixel screen (horizontally) would take 512 iterations through a loop. Even if it's a fast loop, you're talking about a lot of CPU cycles.

When you consider that a lot of lines may get dropped in clipping, then the overall time spent transforming, etc... may be equal to the time spent drawing lines.

My experience has been in optimizing both the 3D and line-drawing algorithms and code as well as I could, and processing a lot of points (about 70% of which ended up on the screen). I spent near equal time doing each.

I suggest reading a computer graphics text for a discussion of line drawing on bit-mapped displays. Remember that you don't want any multiplies or divides in the loop.

Generally, most good line-drawing algorithms on the Mac work something like this **(non-techies can skip this)**:

Given two endpoints P1 (x1,y1) and P2 (x2,y2), decide whether the slope is more vertical, or more horizontal. If $\text{abs}(x1-x2) > \text{abs}(y1-y2)$ then it's more horizontal.

In this case, we want to iterate on x, moving (let's say) from left to right. We iterate on x in this case because we know that y can vary by at most one pixel (the slope is ≤ 45).

Inside the loop, we just have to decide for each pixel in x whether y gets incremented or not. A good way to do this is with a toolbox call to FixedRatio, with which can compute the fixed-point ratio of $\Delta Y / \Delta X$. Ignoring the most significant word of the 32 bit result, we can use the fractional lower 16 bits to add in each iteration of the loop. When a carry is generated, we know to increment Y.

This procedure can be easily generalized for the remaining cases.

Gregory Simons' programming credits include Downhill Racer for Miles Computing, and Smash Hit Racquetball for Primera Software.



Programming in PostScript

© 1986, Denis G. Pelli

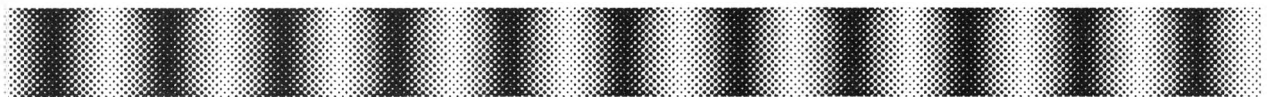
PostScript is a page description language. To print anything on a PostScript-driven printer, such as the Apple LaserWriter, you must send a PostScript program which, when run, will create the image. However, most people who use the Apple LaserWriter have never seen PostScript, because their application, e.g. *MacWrite* or *MacDraw* (via the Macintosh Print Manager), generates and sends the necessary PostScript automatically, when the user says, "Print." PostScript was intended to be used in this way, generated automatically by software, not written by hand. However, there are two reasons why you might want to program in PostScript anyway.

Firstly, PostScript is a powerful graphics language containing lots of powerful elegant operators for doing graphics. In fact, many of its features are not being used by the presently available Macintosh applications, and you might want to get at them. For example, PostScript makes it easy to print rotated and linearly transformed characters, and has powerful operators for halftoning, allowing representation of arbitrary continuous tone images. This newsletter will undoubtedly be festooned with PostScript creations of fellow BMUGers. I suspect that this reason for PostScript programming will only be temporary, as new Macintosh applications will appear to take advantage of more of the features of PostScript.

Secondly, PostScript lends itself naturally to a mathematical description of an image. This is contrary to the current What You See Is What You Get (WYSIWYG) trend in graphics, where you begin with an image in your mind of what you want and fiddle until you get it on screen. Instead, PostScript is best for those occasions where you begin by knowing only the mathematical description of an image, and want to find out what the image would look like. For example, what does a sinewave grating look like?

$$\text{reflectance}(x) = 0.5 + 0.5 \sin(x)$$

It's easy to write a short program in PostScript to produce a good rendition of this,



or any other formula.

Of course, this approach is old hat to anyone who has done computer graphics on a digital video frame store. However, if you want to publish an image, even if you already have a digital video frame store, you still need to get your image onto paper. Unfortunately it is still difficult and expensive to obtain a faithful reproduction on paper of what's on the screen. The biggest problem is the nonlinearity and variability of photography. PostScript allows you to get faithful renditions directly on paper. And you can get a draft from your LaserWriter in a few minutes.

An added attraction of PostScript is the availability of high resolution printers (especially the Linotronic L300), just a modem transfer away, which will run the same program you ran on the LaserWriter, but yield an image at 2540 dots/inch instead of 300 dots/inch.

Okay, so now you're convinced that you must try PostScript. What do you need? First you'll need the *PostScript Language Reference Manual* and the *PostScript Language Tutorial and Cookbook*, both written by Adobe Systems, the company that invented PostScript, and put it in the public domain. (Ordering information for all items mentioned appears at the end of this article.) These two books are very clear and well organized, and are models for how manuals should be written, with a clear separation of the tutorial and reference functions. The tutorial has lots of real examples and the reference manual is organized alphabetically. The only sensible way to do it, but rare nonetheless.

However, these two books only describe the language in the abstract. They barely touch on the practical details of getting your program into the LaserWriter (or other PostScript printer) in the first place. The books only barely acknowledge the fact that in the real world PostScript printers have a finite amount of memory (barely adequate in fact), and that users are only willing to wait a finite amount of time for their image to come out of the printer.

Besides the books you'll need a micro with a plain text editor (to write your program) and connection to a PostScript printer, probably an Apple LaserWriter. Of course if you're reading this article you'll be using a Macintosh, which makes things easier.

There is an odd conspiracy of silence between Apple and Adobe. Apple never really mentions PostScript anywhere (except that they include an out-of-date PostScript manual in their Inside LaserWriter manual), and Adobe never really owns up to the fact that if you're using PostScript the odds are very high that you're programming on a Mac and sending your stuff to a LaserWriter over AppleTalk. (In fact, according to Adobe's recent Prospectus, 82% of their income comes from royalties based on use of their PostScript interpreter in the Apple LaserWriter and LaserWriter Plus.) As a result there are a couple things you really need to know that don't seem to be written down anywhere.

Before I forget, in case you're interested, there's a free PostScript newsletter from Adobe, *Colophon* (which I haven't seen yet), and a column by Don Lancaster in *Computer Shopper*. Their addresses are at the end of this article.

In the rest of this article I will try to pass on what you need to know to program in PostScript on a Macintosh, but which Adobe and Apple won't tell you (at least not in their books).

Talking to Your Printer

The simplest way to connect any computer to the Apple LaserWriter is by a serial line, using any simple communication program, e.g. *FreeTerm*. However, most of us do not own our own LaserWriter, and must share it over an AppleTalk network. Therefore you need a program, a *downloader*, to send your PostScript program (a text file) via AppleTalk to the LaserWriter. There are now lots of such programs around for the Macintosh. Most are free, e.g. *PSDump* 2.5 by A. Oppenheimer and B. Holt, and *PostScript Tester* by Clayton M. Elwell, available from BMUG. However, the best by far of the free downloaders is *SendPS* 1.0 from Adobe Systems. *SendPS* will send your program over AppleTalk to the printer. It will also collect any messages from the printer, sent back over AppleTalk, show them to you as they arrive, and automatically store them in a file for your later perusal. You can also ask for a report on the status of the printer at any time, and you can initiate a remote reset of the printer. *SendPS* doesn't include a text editor, so you'll need one. I recommend *MockWrite*.



I should mention a commercial program that isn't sold as a PostScript downloader, though it does that very well. *JustText* version 1.0i, from Knowledge Engineering, has most of the features of *SendPS* (except showing messages as they arrive), plus a built-in multi-window editor. However, it costs \$195. That is because it also incorporates a powerful page formatter to do typography. *JustText* works with plain text files incorporating special codes surrounded by braces. Naturally the text file that you're working on looks like hell, but the codes and the *JustText* formatter make up for that (somewhat) by offering you more power than any of the WYSIWYGs around at the moment, e.g. automatic hyphenation (rule-based with an exceptions dictionary) and pair-wise character kerning. (Personally, for this flavor of text formatting, I would be more inclined to use *T_EX*, which is much more powerful and is truly device independent—supporting most computers and more than 40 printers—and is now available on the Mac in two independent implementations (See other articles in this newsletter). If this appeals to you, read *The T_EXbook*, by D.E. Knuth who created *T_EX*, and put it in the public domain.) For the PostScript programmer, *JustText* offers the convenience of a built in multi-window editor, and a set of tools (The *LaserTools*™) that come with it. These tools allow you to convert *MacPaint*, *Thunderscan*, and *MacVision* files into pure PostScript.

A common deficiency of both *SendPS* and *JustText* is that they don't tell you when they've finished downloading your program and are just waiting for output. You may know that the job takes a long time, but sends no messages, and it would be nice to know when you can "cancel" (press Command-.) to recover control of the Macintosh without killing your job on the *LaserWriter*.

Receiving the printer messages via AppleTalk is very important, as otherwise you won't know that your program crashed (yes, PostScript programmers make mistakes too) except by the fact that no output came out of the *LaserWriter*. I debugged my first few PostScript programs that way, running to the other room, waiting to see if I got any output, usually didn't, and walking back to edit my program a bit, and try again. It's reminiscent of using the toggles on a PDP-11/20 to program in machine code. Obviously something best left to Real Programmers, as no sane person would want to do it.

An intermediate solution is to use Adobe's free *ehandler.ps* 2.1, an "error break page handler," which is a PostScript program you send to the printer once. From then on (until you reset the printer), when any program crashes you'll get a printed page showing the error message, the offending command, and a peek at what's on the stack. (PostScript is a stack oriented language, like Forth.) However, this still requires running back and forth between your Mac and the *LaserWriter*.

It turns out that the *LaserWriter* (and the *Linotronic* L100 and L300) automatically send the error messages back, via AppleTalk, to the source (your Mac), but apparently it is either too difficult or impossible to get the Macintosh Print Manager to receive this stuff. As a result it seems that the only programs that will receive the stuff from the *LaserWriter* all bypass the Macintosh Print Manager. This seems to take a lot of code (presumably to handle AppleTalk) as these programs are much bigger than the ones that don't bypass the Print Manager. (At MacWorld Expo in Boston last August I think I heard some mumbling about Apple adding two-way communication here.)

Programming Tips

You will definitely want to use either *SendPS* or *JustText*, to get the back talk from your printer. Once you're doing that, you can take advantage of the great "print" commands built into



BMUG FALL 1986

PostScript which are very helpful in debugging. "Print" commands in PostScript do *not* put ink on paper, instead they send a message back to the host. To put ink on paper you say **showpage**. Intuitive, isn't it? Anyway, you get used to it. One especially useful "print" command is "==" which will "print" whatever is on the top of the stack. What's great about that is that *anything* can be put on the stack, even a compiled procedure, and == will uncompile it and "print" it out for you. It took me a while to realize just how useful "printing" is, since the Adobe books don't say anything about debugging.

By the way, the error messages from PostScript are quite specific and helpful, particularly because all the PostScript operators do extensive type and range checking of their arguments and spit out a specific error message if they don't like them. As a result you usually get an error message that directs you to your error, rather than some general sort of machine crash because it tried to do something unreasonable. I still remember when my PL/1 programs on the IBM System 360 model 91 would generate a OCO error, which meant imprecise processor error (i.e. a hardware trap from the cpu somewhere in the pipeline), and whose cause could only be diagnosed by poring over vast core dumps. I haven't done it yet, but there is a way to intercept the PostScript error handlers. Look up "**stop**" and "**stopped**" in the Adobe books. (That's how *ehandler.ps* works.)

When you send your first PostScript program to a LaserWriter (or Linotronic L100 or L300), chances are that someone (maybe you) has previously printed on the LaserWriter using the Macintosh Print Manager. This is significant because the first thing the Macintosh Print Manager does is to make sure that the LaserWriter has a *permanently* installed "Macintosh" dictionary called **md** (that's what is in *Laser Prep* in your System Folder) to interpret a whole bunch of special PostScript calls that correspond more or less to QuickDraw. This dictionary takes up about 60K bytes in the LaserWriter and stays there until you reset the LaserWriter.

The **md** dictionary also creates new versions of the LaserWriter fonts. This is because, in their wisdom, Apple and Adobe chose different *encodings*, that is, assignments of character codes to actual characters in the character set. As I understand it, they mostly agree in the first 127 characters (except for the single quote: ') but mostly differ in their assignments of the remainder of the 255 character codes. Thus the alphabet and usual punctuation are mostly the same, but option characters are different. The Adobe fonts all have readable names, such as Helvetica. The new fonts (differing only in their encoding) created by LaserPrep all have the prefix | ____ (a vertical bar and six underline characters), e.g. | ____ Helvetica. Thus the character code produced by pressing Option-a will produce nothing in Helvetica and will produce "à" in | ____ Helvetica. However, some of the option characters (e.g. | ____ ∂ © Δ μ π ® √ Σ ≈ Ω) are actually stolen from the Symbol font and do not exist in either Helvetica or | ____ Helvetica. To get these characters you should use the | ____ Symbol font.

I imagine that by now everyone knows that the PostScript output of regular Macintosh applications can be intercepted. Use **Chooser** to select the LaserWriter. Within your application select **Print**, and hit **Okay**. Now immediately press and hold Command-F. The PostScript output generated by your program will be put in a new file called "PostScript" instead of being sent to the LaserWriter. (It's a good idea to rename this file right away, before you overwrite it.) This is a plain text file which you can edit with *MockWrite*. Unfortunately it's hard to understand because it uses the procedures defined in the Macintosh dictionary **md**. If instead of Command-F you press and hold Command-K, then the resulting PostScript file will also include the Macintosh dictionary as a header.

There's a new shareware program called *Post-A-Matic* which takes your Apple-generated



PostScript (e.g. from *MacDraw*, captured by Command-F) and translates it to real PostScript, which doesn't require the Apple dictionary. *Post-A-Matic* doesn't do everything, but it's still impressive.

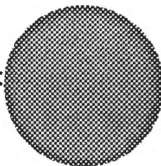
News from BIX!

The LaserWriter driver version 3.1 supports inclusion of raw PostScript in almost anything you send to the printer. Any text whose font is called "PostScript Escape" will be assumed to be PostScript. You can create such a font by using *Font/DA Mover* 3.2 to copy any existing font into a new font file, and using *ResEdit* to rename it. Open the new font file with *ResEdit*. Open the FOND resource, select the resource in that window, and do a **Get Info**. Change the name to *PostScript Escape*. Then close everything. Install the font in your system, using *Font/DA Mover* 3.2. (It will automatically resolve any conflicts arising from having two fonts with the same ID #.) To try it, go into *MacWrite*, and in the middle of some document insert some PostScript, e.g.

```
30 0 rmoveto currentpoint 30 0 360 arc 0.5 setgray fill
```

and set its font to be *PostScript Escape*.

Then print your document on the LaserWriter:



Halftone

One of the more interesting things to do in PostScript is halftoning. PostScript includes an operator called *image* which does all the hard parts, making your job very easy, essentially just to supply the desired gray levels. For example, here's the complete program for the sinewave grating shown on the first page of this article.

```
20 20 rmoveto currentpoint translate
37.5 currentscreen 3 -1 roll pop setscreen
/S 262 string def
0 1 261{S exch dup 16 mul sin 0.9 mul 1 add 127.5 mul cvi put}for
262 1 8 [-0.52084 0 0 0.03 0 0] {S} image
```

Admittedly the Forth-like postfix syntax is hard to read, but it's easy to get the hang of it if you only have to write a few lines of code. The Adobe books make learning and using the language quite easy.

Digital halftoning involves a trade off between spatial resolution (halftone cells per inch) and graylevel resolution (how many distinct gray levels you can produce). Obviously you'd like both types of resolution to be high enough to give the illusion of continuity, without breaks. On the LaserWriter this is usually not quite achievable, and you will want to optimize the tradeoff for your particular image.

If the spatial resolution is too low, you will be able to resolve the halftone cells, and your image will be seen to be made up of dots. Note that this depends on viewing distance; you can always view your image from far enough away that your eye will be unable to resolve the dots. At a typical reading distance of 16 inches you would require about 150 halftone cells per inch to make the halftone screen nearly invisible.

If the graylevel resolution is too low, you will see steps in graylevel where there ought to be only

gradual changes. For example, a closeup photo of a face typically has gradual graylevel changes across the cheeks, and contour lines will appear if the graylevel resolution is too low. The maximum graylevel resolution attainable using the halftone operators in PostScript is 257 levels, which will be enough for most images. (The steps near black may still be visible.) If your criterion is that the resultant image be indistinguishable from an image produced with an infinite resolution of graylevel, then even 257 levels is not enough for many images. However, most images, seen on their own, will look all right with only 65 levels of gray.

65 levels of gray requires a halftone cell with at least 64 pixels, or a square of 8×8 pixels. (The 65 levels correspond to number of pixels, out of 64 in the halftone cell, which are white: 0, 1, ... 64.) On the LaserWriter, with 300 dots/inch, this will produce $300/8=37.5$ halftone cells per inch, which is a bit low. That's what I used for the sinewave example on the first page of this article. Alternatively, with 150 halftone cells per inch, you will have only $2 \times 2=4$ pixels per halftone cell, and thus only that many levels of gray (plus 1). The default on the LaserWriter is inbetween: 60 halftone cells per inch and $5 \times 5=25$ levels of gray (plus 1). The gray disk above has 60 halftone cells per inch. The best compromise depends on the particular image. Use the PostScript **currentscreen** and **setscreen** operators to experiment.

A Halftone Bug in PostScript

Incidentally, you should know that there is a bug in the PostScript halftoning software which produce systematic distortions in the rendered gray levels unless you have exactly 256 pixels in each halftone cell.

To discuss this in detail it will be helpful to use the nomenclature of the *Adobe PostScript Language Reference Manual*, pages 80-81. Each cell of the halftone screen is made up of n pixels. The user will specify a desired gray level g , and PostScript will set i of the n pixels to white, setting the others to black. Thus the user will request a gray level g and obtain a gray level i/n . One would like i/n to be as close as possible to g . The question is, how does i depend on g and n ?

The formula, $i = \text{floor}(g \times n)$, given on page 81 of the manual, is wrong.

By various experiments I discovered that g , the requested gray level, is quantized to 257 levels, from 0 to 256:

$$G = \text{floor}(256 \times g + 0.5) \quad (1)$$

These experiments indicate that the formula for i used by PostScript version 38.0 is as follows

$$\begin{aligned} &= 0 && \text{if } G=0 \\ i &= \text{floor}((n/256) \times (G + 0.5)) && \text{if } 0 < G < 256 \\ &= n && \text{if } G=256 \end{aligned} \quad (2)$$

However, even accepting the need to quantize g (presumably to economize on memory), this formula does not choose the value of i which would produce a gray level i/n as close as possible to the requested gray level g .

The correct formula, for most accurate rendering, would be

$$i' = \text{floor}((n/256) \times G + 0.5) \quad (3)$$



Note that the two formulas appear very similar. The only difference is that in formula (2) the 0.5 is added before the multiplication and in formula (3) it is added after the multiplication. This difference produces noticeable errors in the rendering of halftones.

What are the consequences of PostScript's use of i (from formula 2) instead of i' (from formula 3)? It is helpful to slightly rearrange the formula for i :

$$\begin{aligned} &= 0 && \text{if } G=0 \\ i &= \text{floor}((n/256) \times G + (n/256) \times 0.5) && \text{if } 0 < G < 256 \\ &= n && \text{if } G=256 \end{aligned} \quad (4)$$

Note that when n equals 256 the quantity $(n/256) \times 0.5$ will equal 0.5, so i and i' will be equal. However, when n is smaller than 256 then i will be too small, by as much as one pixel. When n is larger than 256, then i will be too large, by as much as $n/512$ pixels.

The smallest value of n is 1, a one-pixel cell. In this case i is too small, and in fact is zero, producing black, unless $G=256$ (i.e. $g \geq 255.5/256 = 0.99805$). At the opposite extreme, if n is 1024 (a 32×32 pixel cell, e.g. a screen frequency of 9.375/inch on the LaserWriter) then i is too high by about $1024/512 = 2$ pixels, making all the gray levels about $2/1024 = 0.002$ lighter than requested.

When $n=256$ the formulas become particularly simple:

$$\begin{aligned} i = i' &= \text{floor}(G + 0.5) && \text{since } n=256 \\ &= G && \text{since } G \text{ is an integer} \\ &= \text{floor}(n \times g + 0.5) && \text{by (1), since } n=256 \end{aligned} \quad (5)$$

Halftone rendering will be as accurate as possible only when $n=256$. Grays will be too dark when $n < 256$, and grays will be too light when $n > 256$.

Other PostScript Printers

Lots of other companies are bringing out PostScript compatible printers, mostly with 300 dots/inch resolution, same as the Apple LaserWriter. I've heard of a third-party upgrade for the HP LaserJet to make it PostScript compatible.

Linotype makes two high resolution PostScript-compatible printers, the Linotronic L100 (\$35,000) with a maximum resolution of 1270 dots/inch and the Linotronic L300 (\$60,000) with a maximum resolution of 2540 dots/inch. Linotype calls them its "Series 100" printers. Oddly enough, the resolution of these computerized printers must be set *manually*, to 615, 1270, or (on the L300), 2540 dots/inch.

The 300 dot/inch LaserWriter is fine for drafts, and the 2540 dots/inch Linotronic L300 will produce excellent final prints, on either Kodak RC (i.e. Resin Coated) paper or film. My experience is that the Linotronic L100 is useless for halftoning because any gray areas that you print will have dark lines every inch or so down the page. Apparently this is a result of the many stops and starts during printing, and the poor paper handling of the L100 when compared with the L300. (I'm not sure whether the dark lines are due to larger dots or greater dot density. In any event it's only a problem for halftones and bit maps. Solid black and white areas, even fine text, are not visibly affected.) Prices for using the L100 and L300 are very similar, about \$10 a page, so use the L300.

Here are seven shops which will run PostScript jobs on Linotron L300 printers. All will accept



job submissions by modem or on a Mac disk. They typically charge \$10 per page (plus postage), and try for, but do not promise, one-day turnaround. I always ask to have my printout sent by Federal Express. (Source: *Personal Publishing*, July, 1986.) (I've only used MacTypeNet, and am happy with their service, though they don't always achieve one-day turnaround.)

Cobb Typesetting	Cincinnati, OH	(513)-241-1146
Design North, Inc.	Racine, WI	(414)-762-1320
Desktop Publishing Inc.	San Rafael, CA	(415)-258-0767
MacTypeNet	Livonia, MI	(313)-477-2733
National Colorite Corp.	New Berlin, WI	(414)-784-8980
ProTypography	Chicago, IL	(312)-266-8973
Typesetting Services Corp.	Providence, RI	(401)-421-2264

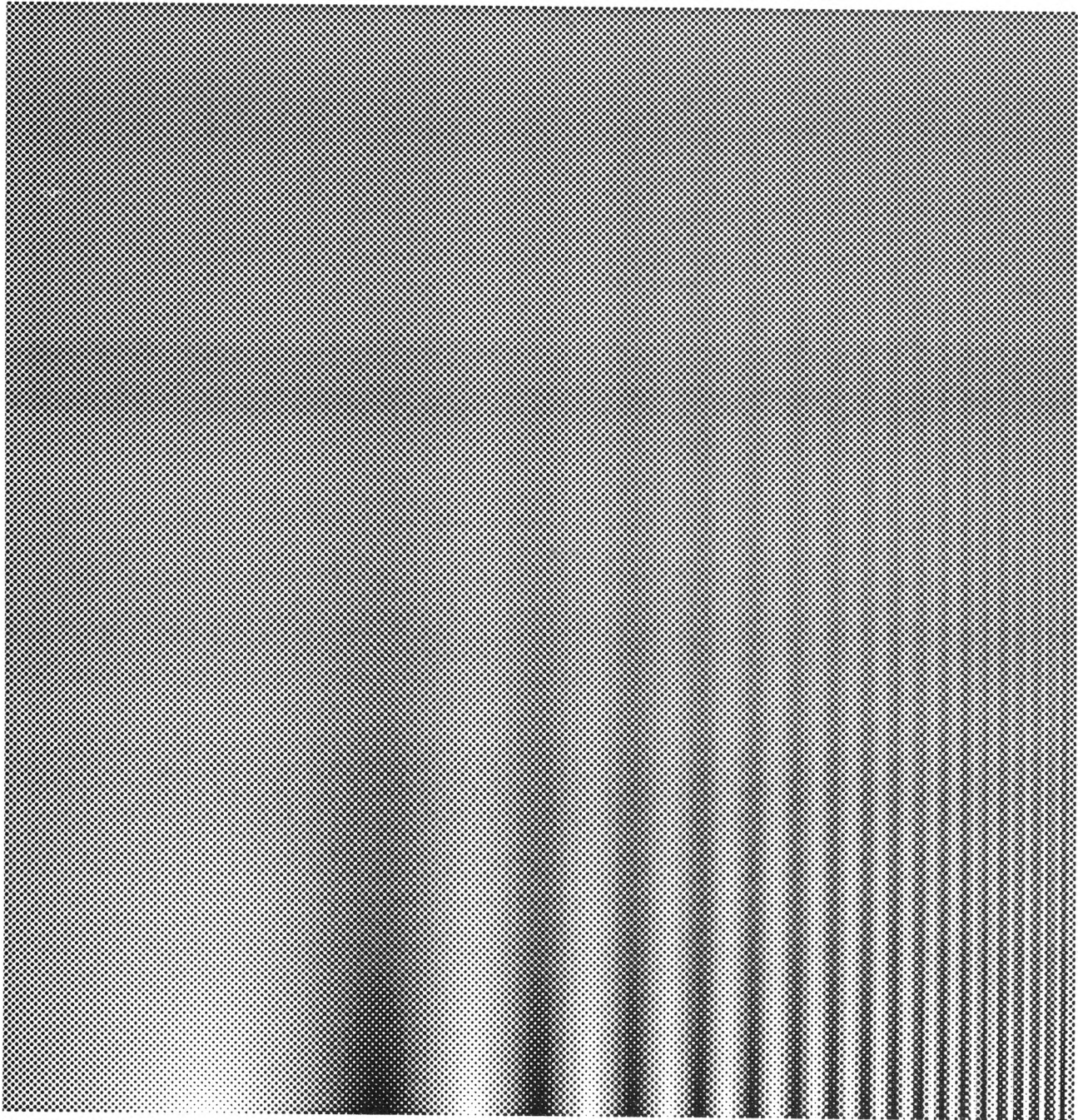
Some Linotronic print shops will charge extra or refuse to run jobs that take many hours. However, the Adobe books give no information about timing. I find that I can get a reasonable estimate of runtime on the Linotronic by first measuring how long the job takes on the LaserWriter and multiplying by the factor by which the number of pixels will be increased. E.g. if the job takes 1 minute at 300/inch on the LaserWriter then it will take $(2540/300)^2 = 72$ times as long, i.e. 72 minutes, at 2540 dots/inch on the L300. This is because the LaserWriter (and the Plus) and the Linotronic L100 and L300 all use very similar processor boards, using the 68000 processor. However, I've just learned from Adobe that by December, 1986, Linotype will have upgraded the ROMs in all the Linotype L100 and L300 printers from PostScript version 38.0 up to 39.0. Apparently this speeds up printing a lot, ranging from a factor of 1.5 times faster for printing of non-cached fonts to nearly a factor of 20 for bit-mapped images. Apparently Apple is *not* planning to offer an upgrade of the LaserWriter ROM to version 39.0. (Adobe has just announced a 68020 "Atlas" board which is supposed to run 3 to 4 times faster than the 68000 boards, but I don't know if it's compatible with the LaserWriter or Linotronics.)

An odd deficiency of PostScript, given that it is a device-independent page-description language, is that it lacks any general device-independent way to describe the size of the page. To obtain nonstandard page sizes on the Linotronic printers (up to 11.7" by 25.7" at 1270 dots/inch, less at 2540 dots/inch) you need to use a special operator, **setpageparams**, which is documented in a Supplementary manual for the Linotronic printers (see below).

An Example

The image below shows a sinewave whose frequency increases horizontally as 10^x , and whose contrast falls vertically as 10^{-y} . At each frequency if you follow the bars up with your eye, you will find that eventually they disappear. The more sensitive you are to that frequency the less contrast you need to see it, and thus the taller those bars will be. As a result, the outline of visibility which you see, which typically has an inverted U shape, describes your *contrast sensitivity* as a function of spatial frequency. This pattern was originally invented by Campbell and Robson.

BMUG
Fall 1986





Items Mentioned

PostScript Language Tutorial and Cookbook by Adobe, "The Blue Book", Addison Wesley, 1985, (\$16.95)

PostScript Language Reference Manual by Adobe, "The Red Book", Addison Wesley, 1985, (\$22.95)

PostScript Supplement for the Linotype Series 100, Version 38.0, by Adobe, 1985, Revised May 15, 1986. Adobe Systems Inc., 1870 Embarcadero Road, Palo Alto, CA 94303, (415)-852-0271

Colophon, PostScript newsletter (free) Adobe Systems Inc., 1870 Embarcadero Road, Palo Alto, CA 94303, (415)-852-0271

Computer Shopper (\$21/year) The Ask the Guru column by Don Lancaster has occasional tidbits on PostScript. Computer Shopper, P.O. Box 1419, Titusville, FL 32781

SendPS 1.0 program, included in the *COLOPHON 2 IMAGES* disk from Adobe (\$12 including postage & handling). Adobe invites free distribution of SendPS, but forbids selling it without permission. Adobe Systems Inc., 1870 Embarcadero Road, Palo Alto, CA 94303, (415)-852-0271

Font Downloader 2.0 program (\$185) Included on Adobe font disks, which cost about \$185. SendPS is better for program downloading. Adobe Systems Inc., 1870 Embarcadero Road, Palo Alto, CA 94303, (415)-852-0271

PSDump 2.5 program, BMUG PostScript disk

PostScript Tester 1.1 program, renamed "PostScript Worker"

ehandler.ps error break page handler program. Written by Adobe. Current version is 2.1. Both on BMUG disk #31

JustText program (\$195) Knowledge Engineering, G.P.O. Box 2139, New York, NY 10116, phone (212)-473-0095

MockWrite 4.3.2 desk accessory program (\$35 shareware), CE Software, 801 73rd Street, Des Moines, IA 50312, phone (515)-224-1995, Also available on BMUG disk #20, 11, etc.

Post-a-Matic 3.3z (shareware \$50 Canadian), Peter Bennett, Graphic Presentation Charts, 23 Exford Drive, Scarborough, Ontario, CANADA M1P 1L5, phone (416)-751-2727 home, Compuserve 76314,1670

ResEdit and **Font/DA Mover** 3.2 Apple Computer. BMUG disks #10, 18, 20

The T_EXbook by Donald E. Knuth, Addison Wesley, 1984

MacT_EX program (\$750), FTL Systems, 234 Eglinton Ave. East, Suite 205, Toronto, Ontario, CANADA M4P 1K5

MacT_EX 0.5 program, Addison-Wesley (617)-944-6795

Denis Pelli does vision research, and is using PostScript to produce a new kind of vision test chart. He's an Associate Professor at the Institute for Sensory Research, Syracuse University, Syracuse, NY 13244-5290, (315)-423-4164. This research was supported by NIH grant EY04432.



Adventures with PostScript

Arthur Luehrmann

The main business of our company is writing and publishing books about computing—high-school and college textbooks, programming manuals, self-study guides to application software packages, and the like. For a new computer book, time is of the essence: 12 or 15 months spent in the production queue (typical of traditional publishing methods) often means that the book has already lost a third of its useful life before it is born. Naturally enough, we are keenly interested in discovering ways to speed the process without giving up quality.

A decade ago, we broke ground with a wonderfully Rube Goldbergian system. Using a simple ASCII text editor on an early CP/M micro, we entered, proofed, and edited the manuscript, including our own codes to mark the beginnings of chapters, sections, paragraphs, program listings, and other typographic features. Next we ran a program that converted our codes to the ones expected by a Compugraphic 4000 typesetting machine. Communication between micro and typesetter was via—believe it or not—paper tape. Within a day or two of final editing of a manuscript, we would arrive at our friendly neighborhood typesetting shop with a bushel basket of paper tape. Inside another couple of days, we received composed pages for a whole book.

Since then, we have moved with the times and the technology. We've composed several books by sending coded manuscript on disk to shops that have staff programmers ready to convert our high-level codes into low-level codes needed by their typesetting systems. We've done other books entirely via Donald Knuth's $\text{T}_{\text{E}}\text{X}$ software at Stanford. However, all systems we've explored in the past have had one big weakness: no support for graphics. Our books feature hands-on activities and need hundreds of figures showing what the user should see on the screen at each moment.

Which brings me to PostScript. Unlike previous typesetting languages, PostScript is, at bottom, a graphics language. The basic instructions in PostScript tell the computer where to place graphic elements on a page. Type itself is just a graphic element, as are points, lines, curves, and filled areas. So PostScript seemed to be the answer to our number-one problem.

In the rest of this article, I want to show some examples of the sorts of graphics that can be produced by direct programming in PostScript. Then I want to return to our specific problem: creating screen figures as part of the book composition process. Finally, I will say some things about how to get started with PostScript.

Hilbert Curves

If you use a LaserWriter at all, you are using the PostScript language, but you can be blithely unaware of the fact. Your application and the LaserPrep file automatically translate your output into PostScript commands. You never see the PostScript program that results, but the LaserWriter does.

If PostScript is the native language of the LaserWriter (and the Linotronic typesetting machines), can't you just write PostScript programs and send them directly to the LaserWriter? The answer is yes, and you don't even need a Macintosh. Details on how to do this will come later. For now, I'd like to show an example of a complete PostScript program (Fig. 1) and the output it produces on a LaserWriter (Fig. 2).



BMUG FALL 1986

```
%%Title: Hilbert
%%Creator: Arthur Luehrmann
%%CreationDate: 27-Oct-86

/inch { 72 mul } def
/edge 2.2 inch def % edge of level-1 curve
/lw 4 def % line width of level-1 curve

/hilbert % Stack: recursion-level ==> -
{
  dup 0 gt
  {
    1 sub dup dup dup
    -1 1 scale
    90 rotate
    hilbert
    -90 rotate
    -1 1 scale
    0 edge rlineto
    hilbert
    edge 0 rlineto
    hilbert
    0 edge neg rlineto
    -1 1 scale
    -90 rotate
    hilbert
    90 rotate
    -1 1 scale
  }
  pop
} ifelse
} def % end hilbert

%%----- End Prolog -----

/Helvetica findfont 14 scalefont setfont
2.6 inch 8 inch moveto
(Hilbert Curves of Order 1 through 5) show

2 inch 3 inch translate % lower left of figure
1 1 5 % draw 5 hilbert curves
{
  newpath
  lw setlinewidth
  edge 2 div dup moveto
  hilbert
  stroke
  /edge edge 2 div def
  /lw lw 2 div def
} for

showpage
```

Fig. 1. PostScript listing for Program Hilbert

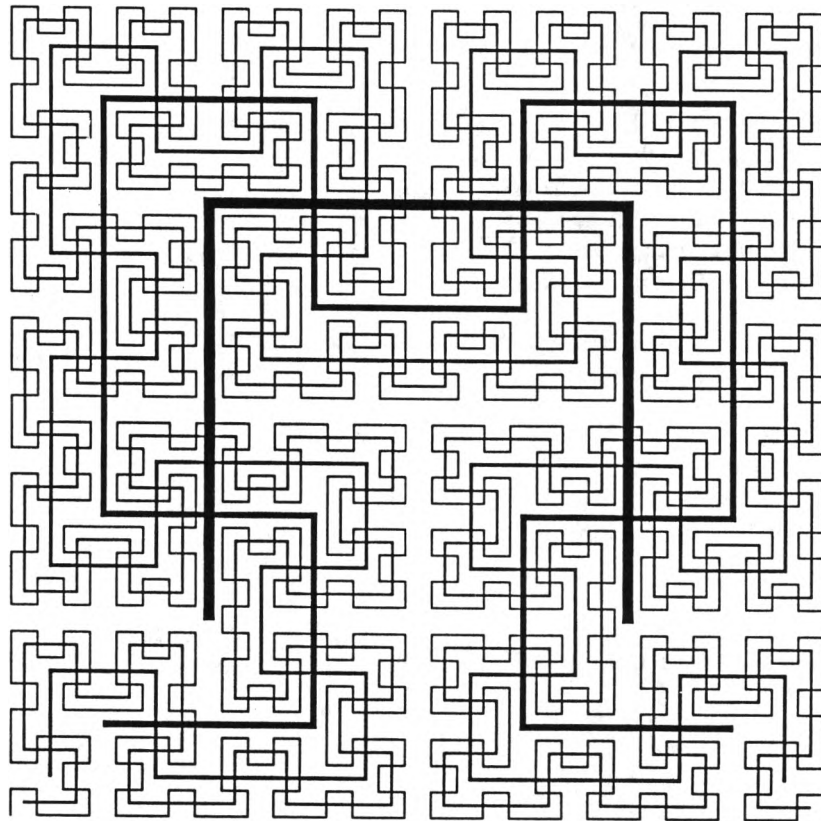


Fig. 2. Hilbert Curves

Program Hilbert draws the famous space-filling curve described long ago by the mathematician of the same name. The "level-1" curve has the thickest lines; "level-2" has lines half as thick; and so on. As the level number goes up, the curve comes closer and closer to filling a square. Computer science types will recognize the recursive nature of this family of curves: each curve is built out of four copies of the curve one level-number below it (suitably rotated and connected). In fact, procedure Hilbert in the program listing contains four calls to itself.

Unless you're a Forth freak, PostScript programs will look bizarre. There is no assignment statement and no way to write expressions such as $2*5+9$. In fact, the way to tell the computer to perform this calculation in PostScript is to write this list of five items, separated by spaces:

```
2 5 mul 9 add
```

Operations are performed by placing items on a stack. In this example, the 2 goes on the top of the stack. Then the 5 goes on top of the 2. Next comes mul, an operator: it tells the computer to remove two numbers from the top of the stack, multiply them, and place the result (10) on top of the stack. Next 9 goes on the stack, on top of the 10. Then comes add, another operator, which tells the computer to remove two numbers from the top of the stack, add them, and place the result (19) on the top of the stack.



As in other programming languages, data items can have names in PostScript. In program Hilbert a line near the beginning is this:

```
/lw 4 def
```

Here is what happens when this three-item list is read by PostScript. First, the name `lw` is put on the stack. Then the number 4 is put on the stack. Next comes `def`, an operator, which tells the computer to remove two items from the stack, put the bottom item (which must be a name) into a list called a dictionary, and associate the top item with that name. Future references to the name `lw` will result in the value 4. In other words, this three-item list has the same effect as "Let `lw = 4`" in BASIC or "`lw := 4`" in Pascal.

The `def` operator is also used in PostScript to associate names with procedures. The first non-comment line in program Hilbert is an example:

```
/inch {72 mul} def
```

This associates the name `inch` with the two-item list "`72 mul`." If at a later point in the program, PostScript finds, for example, "`2.2 inch`," the effect is the same as if it had found "`2.2 72 mul`."

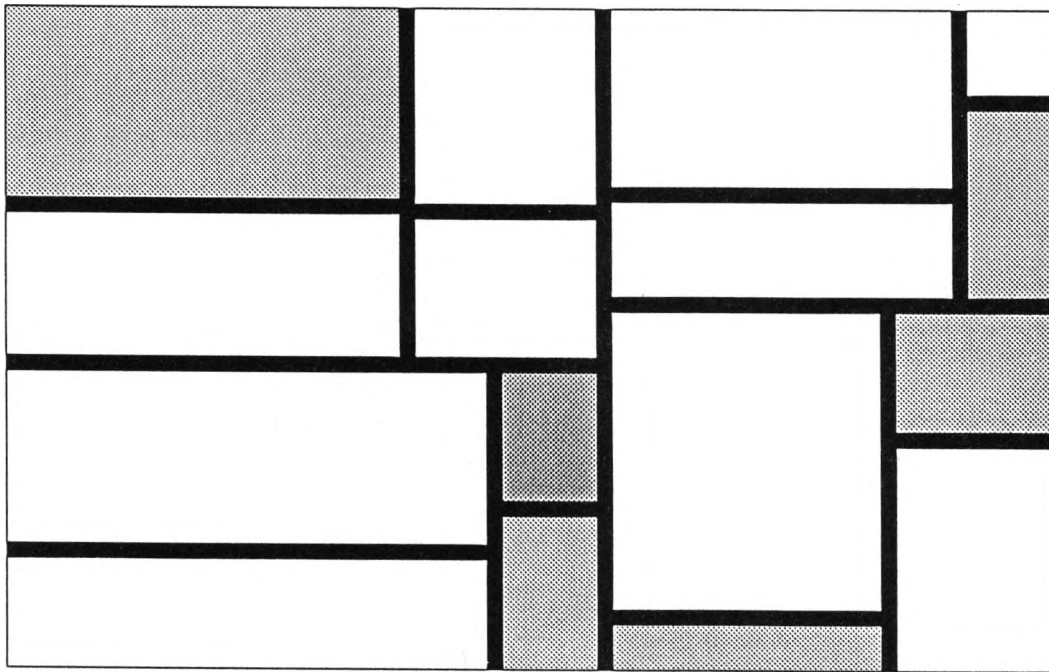
This is not the place for a full-bore primer on the PostScript Language. It is enough now to get the flavor of a PostScript program and to see what sorts of output can be generated by a few fairly simple examples.

Piet Mondrian

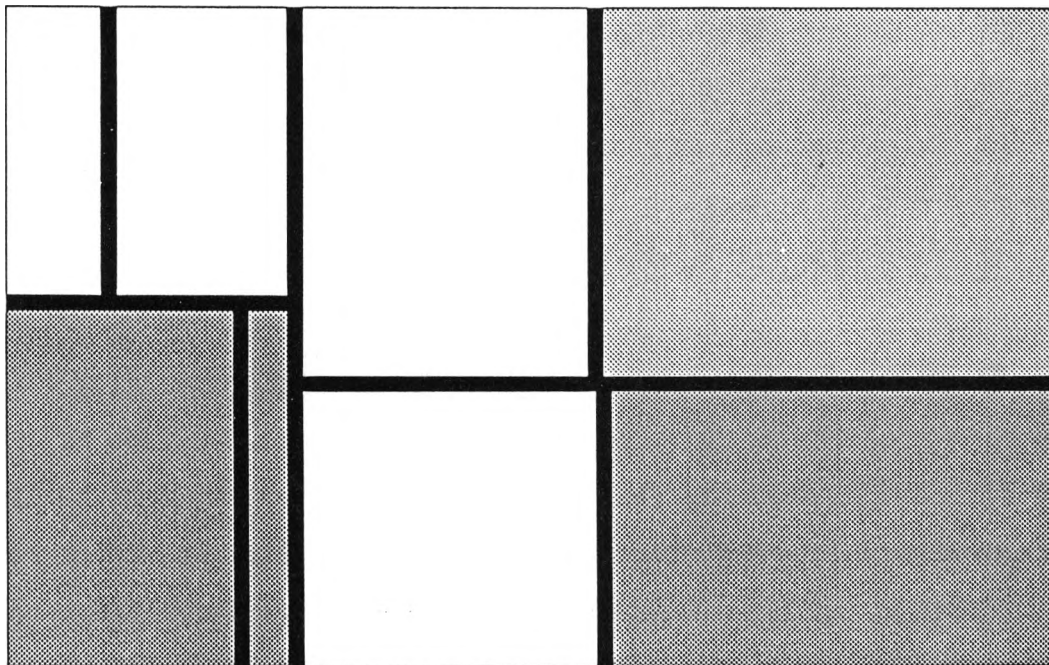
The next two examples (Figs. 3, 4) are also drawn directly from a PostScript program. They represent a crude attempt to generate "art by the yard" in the style of the Dutch modernist, Piet Mondrian. Each drawing produced by the program is different. By choosing random values for positions of lines, the number of lines, whether or not to fill a rectangle with gray, and what shade of gray to use, the program makes it highly unlikely that a drawing will repeat itself.

Color is lacking, of course. However, this is the fault of the LaserWriter, not PostScript. Operators exist for specifying both shades of gray and also mixtures of colors. If the printing device handles colors, PostScript can tell it precise color information on a continuous palette.

Mondrian is another recursive program. Basically, there are two actors here: one guy slices a rectangle vertically into two parts; the other guy slices a rectangle horizontally. At the start, the first guy is given the rectangle that defines the outer border of the picture. After dividing it vertically, he sends the left half to the other guy. Then he sends the right half to the other guy. When the second actor gets a rectangle, he divides it horizontally, and then sends each part back to the first guy. This process goes on for a random number of times. When either guy decides it's time to quit, he makes the other decisions: whether to fill the rectangle just received and, if so, with what shade of gray.



Number of levels = 4



Number of levels = 3

Fig. 3. (Upper) Homage to Piet Mondrian with 4 levels
Fig. 4. (Lower) Homage to Piet Mondrian with 3 levels

Hannes Beckmann

I am particularly fond of the next two examples (Figs. 5, 6). Hannes Beckmann was a Bauhaus artist and dear friend from my professorship days at Dartmouth College. Once as I was having lunch in the performing arts cafeteria, I noticed hanging on the wall some student work from one of Hannes' design classes. All the pieces were constructed out of the same materials: a large black cardboard rectangle upon which were glued hundreds of thin bamboo sticks, all of the same length. The design problem was to see what could be done with limited means, and the variety of solutions was impressive.

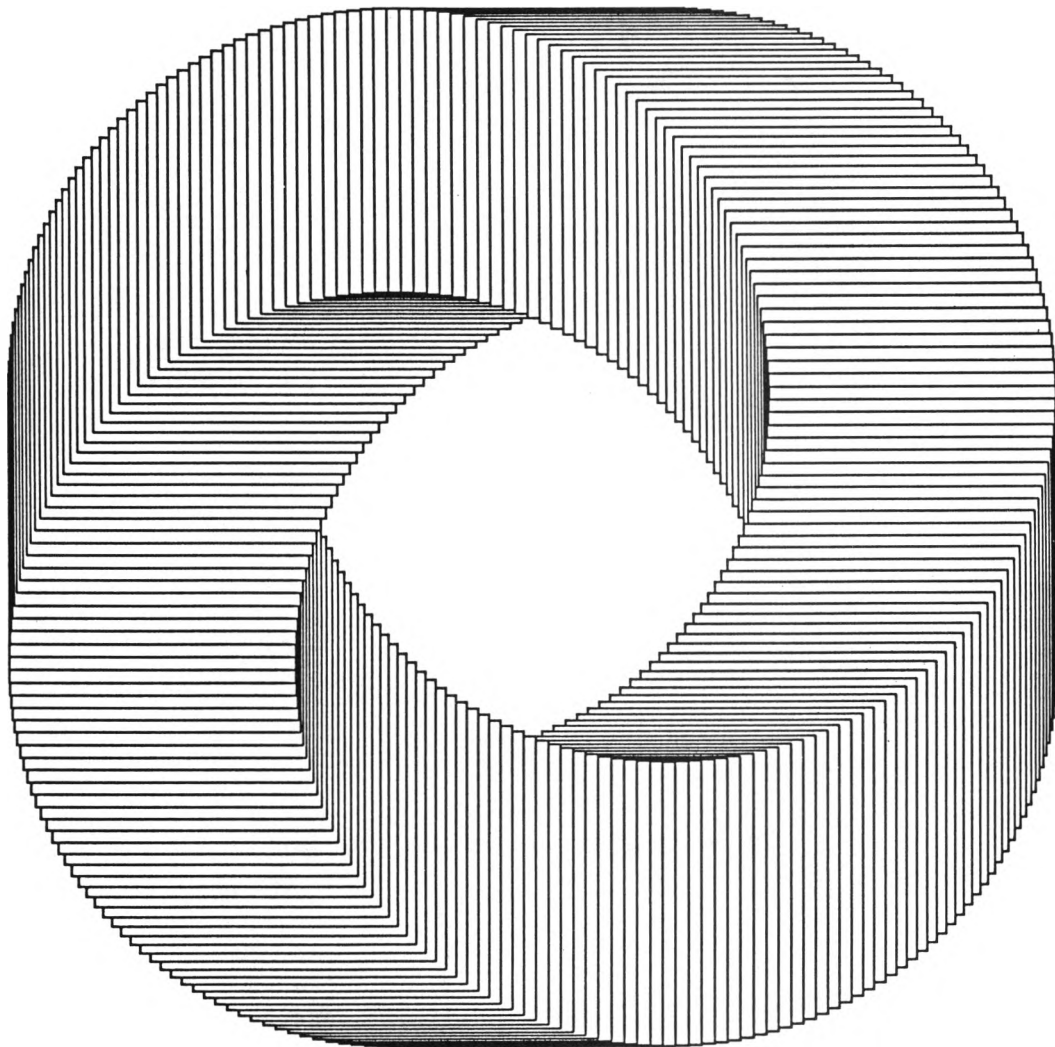


Fig. 5. Homage to Hannes Beckman One



One work amazed me. The sticks defined the edges of overlapping squares. The stack of squares seemed to rise in a spiral. Then, somehow, the top of the spiral joined smoothly with its own bottom— somewhat like Escher's impossible drawings. Hannes later did a series of oils on this same theme. I was captivated by this and spent many hours trying to describe the figure in algorithmic form.

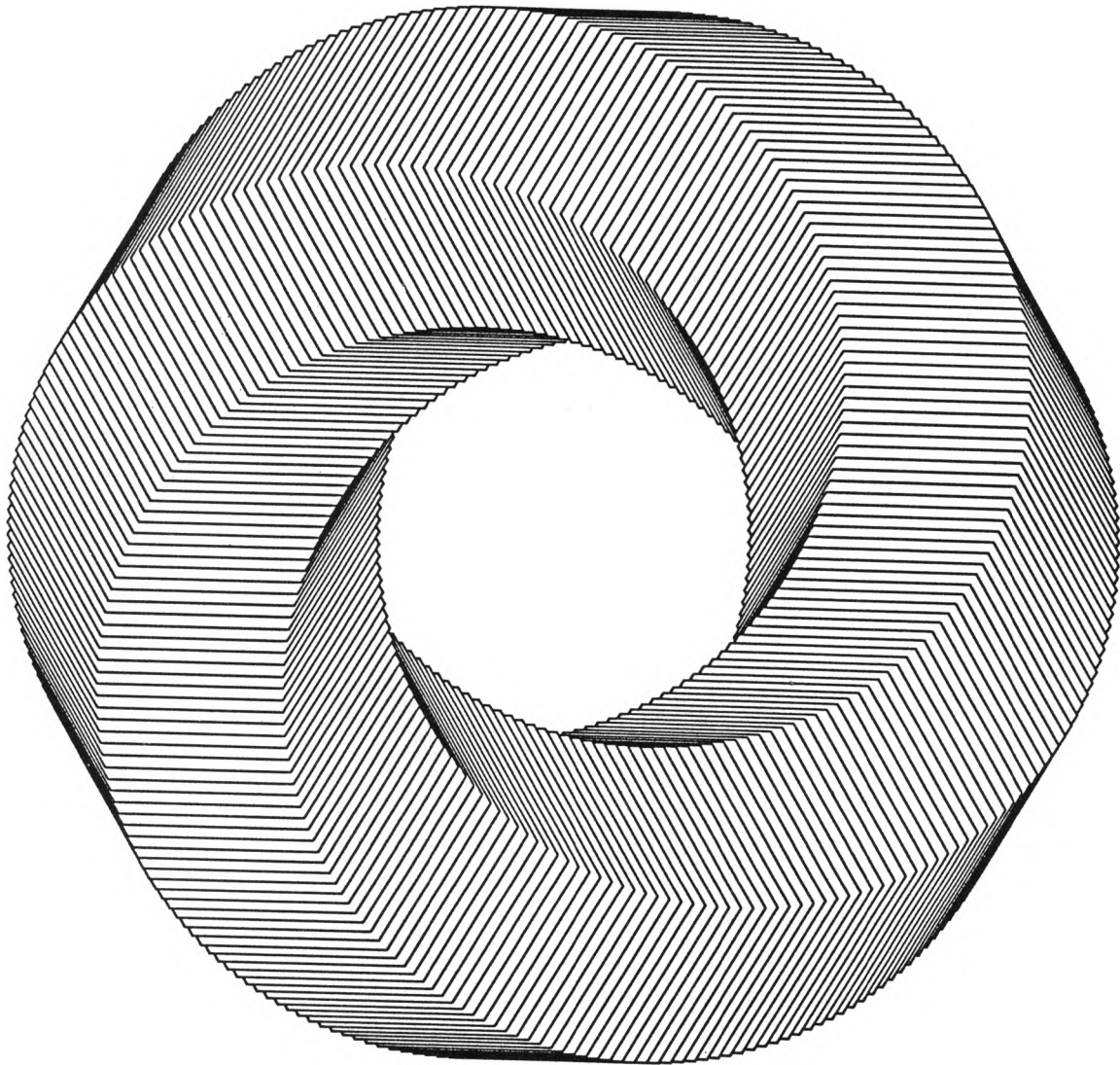


Fig. 6. Homage to Hannes Beckman Two

The result, as often happens in computer work, was a generalization: a program that could reproduce the overlapping squares, but could also draw overlapping triangles, pentagons, hexagons, and so on. The figures here are two examples.

Back to Business

As I said at the outset, our practical interest in PostScript is motivated by the need to create hundreds of high-quality, low-cost screen figures for inclusion in books that feature hands-on computer activities. We especially need a system for creating 80-column text screens such as seen on Apple II and IBM PC computers.

Anyone who has tried to produce 80-column screen figures for publication knows the problems. The results are inaccurate or ugly or both; and, by whatever method used, they are very costly. We hoped that PostScript devices would give us a way to beat all these problems. Preliminary results for a book we're writing on AppleWorks (Figs. 7, 8) are quite encouraging.

Lab 1-1 Screen Figure 1

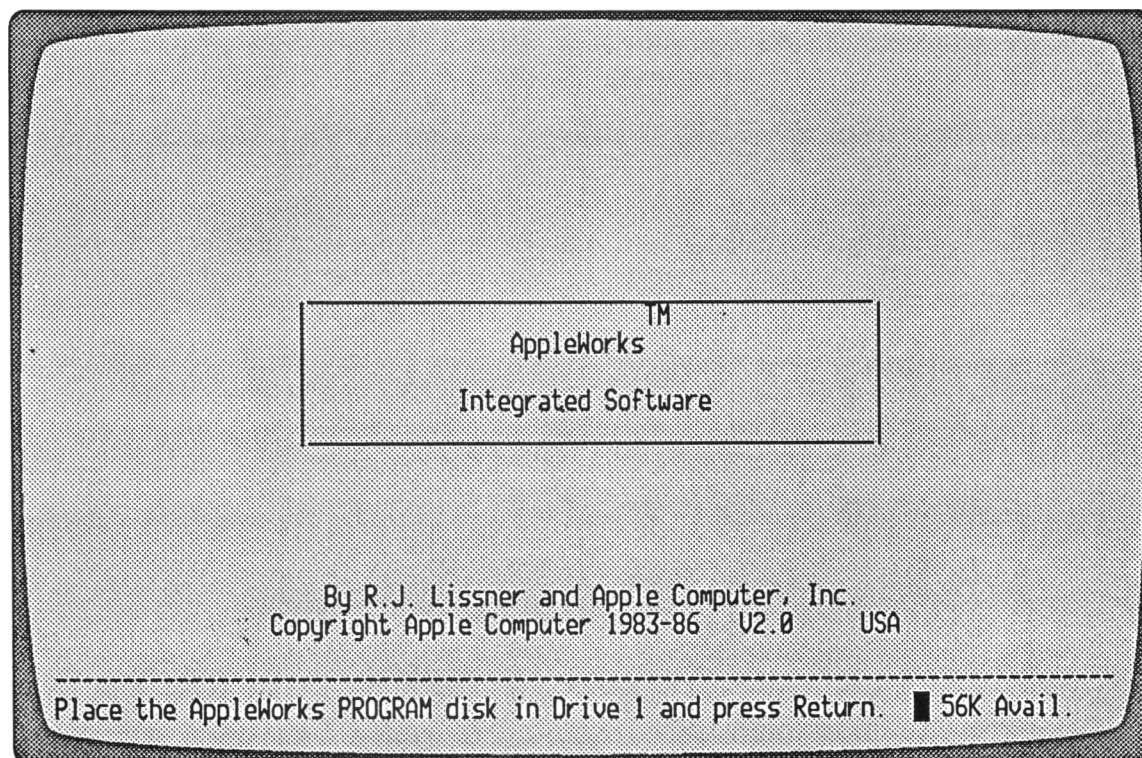


Fig. 7. Opening AppleWorks screen created in PostScript

The system is simple to use. The first step is to acquire a file that contains the text that appears on the screen. Some screen-dump software can make these files quickly and automatically. (If such screen dumps are not available, it takes only a few minutes to type the text by hand and proof it.) The next step is to run a program that does two things: first, send a special set of



PostScript procedures to the PostScript device; second, read the text of each screen figure and send appropriate PostScript commands to draw the text, highlighting, and borders.

Lab 1-1 Screen Figure 4

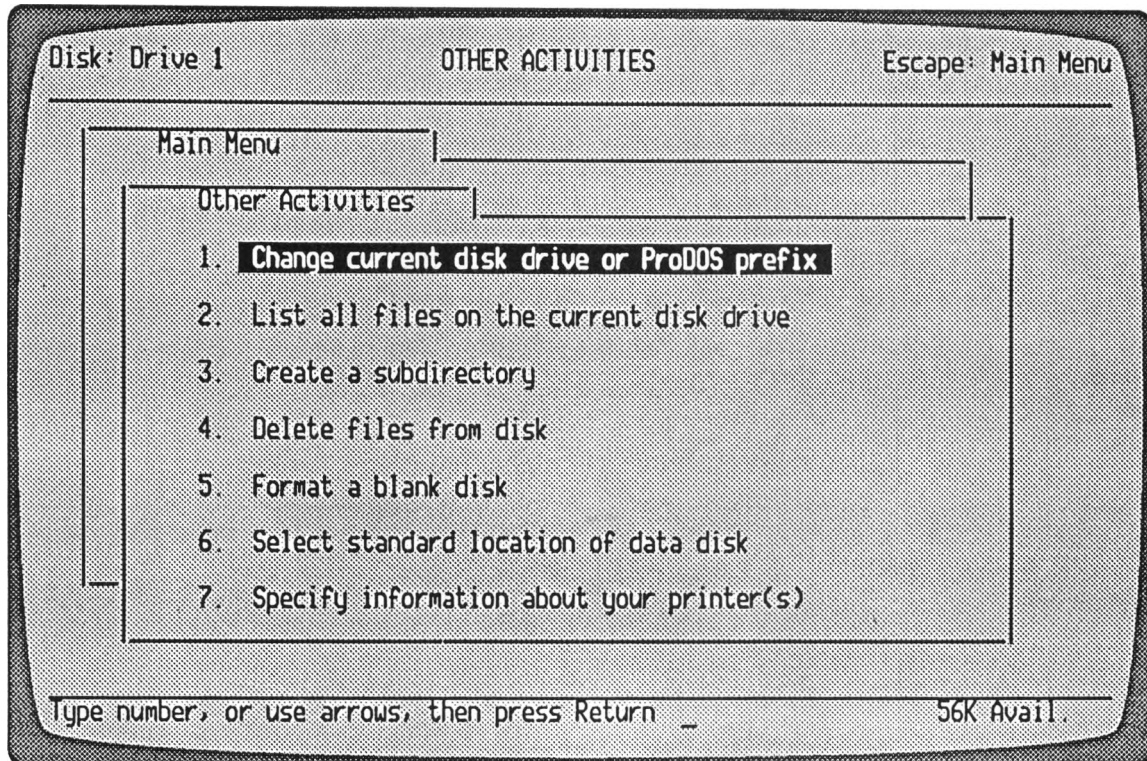


Fig. 8. Other Activities AppleWorks screen created in PostScript

One question that we faced was what typeface to use for the screen text. On 80-column displays, the characters all have the same width. The only available monospaced font today is Courier, a typewriter-style face that bears no resemblance to the dot-matrix faces used on 80-column displays. We tried Helvetica, scaling each character up or down to make it fit, but that also changed the weights of the stems and looked ugly.

Our solution was to create our own laser font (Fig. 9). We used FONtographer for this. It seemed a pity to use this powerful tool, capable of designing beautifully curved typefaces, merely to emulate a basically ugly dot-matrix face, but that was the task at hand. To do this, we set up a grid of guidelines that corresponded to the position of the dots in the Apple II dot matrix. Then we stroked each character by drawing straight lines from point to point on this grid. All in all, it took about a day's work to build the font. The result is what might be called a "stroked dot-matrix" typeface—prettier and easier to read than a true dot-matrix font, but still quite faithful to the original. We named the font AppleDot.



BMUG
FALL 1986

ABCDEFGHIJKLMNOPQRSTUVWXYZ
abcdefghijklmnopqrstuvwxyz
`1234567890-=
~!@#\$%^&*()_+
[]{}|;':",./<>?

24 pt. The quick brown fox jumps over
18 pt. The quick brown fox jumps over the lazy dog.

12 pt. The quick brown fox jumps over the lazy dog. 0123456789

9 pt. The quick brown fox jumps over the lazy dog. 0123456789

6 pt. The quick brown fox jumps over the lazy dog. 0123456789

This print sample produced by Fontographer, the professional font editor
from Altsys Corp.: 720 Ave. F, Suite 108 Plano, TX 74074 (214) 424-4888

Total memory:
332292

Memory used:
141592

Free memory:
190649

Fig. 9. The AppleDot font created with FONtographer

Exploring PostScript

This has been a brief show-and-tell about the merits of working directly with PostScript and bypassing LaserPrep, AldusPrep, and any other Prep. As always with software, the best way to



learn is to noodle around at the computer, preferably with a good book at hand.

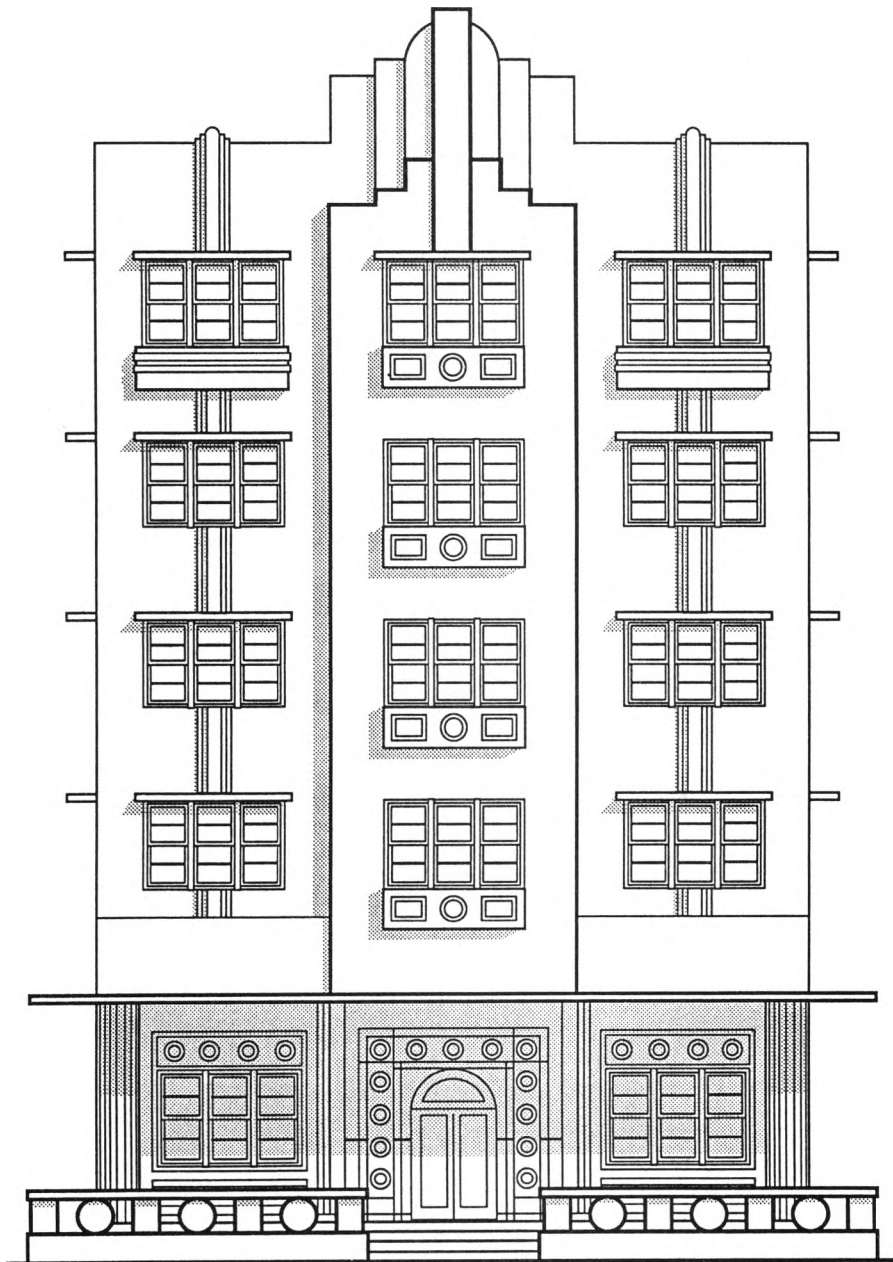
The official Adobe books are your best bet: *PostScript Language Tutorial and Cookbook* and *PostScript Language Reference Manual* (Addison Wesley, 1985). They are extremely well written, thorough, reliable, and well indexed—very far above the average for software documentation.

Unless you are willing to write an emulation program, you do need a LaserWriter, since that is where the PostScript interpreter lives. By using a communications program on any computer, you can run PostScript in an interactive mode, entering a line at a time and getting messages back from the LaserWriter. This works, and you can do the exercises in the Adobe *Tutorial* this way. However, you won't be happy as your programs get longer, since they disappear into the bowels of the LaserWriter as you enter them, never to be seen again.

The ideal system is a combination text editor and file downloader. You create your programs in the editor and then download them into the LaserWriter. The best choice here is *JustText*. (It is actually a typesetting language that generates pure PostScript code and is interesting for that reason alone. However, *JustText* can also be used for the purpose described here.) The download command in *JustText* both sends PostScript files to the LaserWriter and displays on the screen any error messages that come back.

Is the payoff from learning PostScript enough to justify the effort? That depends on your needs. If you are satisfied with the standard Macintosh applications and the way they appear when sent to the LaserWriter, your time is probably better spent on other things. However, if you want full graphic control of the LaserWriter or Linotronic devices, PostScript is the answer.

Arthur Luehrmann, Ph.D., is a partner in Computer Literacy. He formerly taught Physics at Dartmouth and was involved with the Computer Division of the Lawrence Hall of Science. He would be glad to talk with readers who need high-quality screen figures for documentation and manuals. Contact him c/o Computer Literacy, 1466 Grizzly Peak Blvd., Berkeley CA 94708.



An architectural rendering in MacDraw by Ames Kanemoto/Paper Crane Graphics

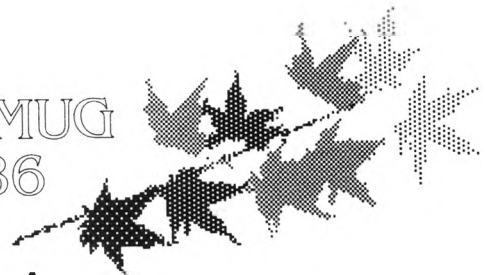
BMUG
Fall 1986



Bits & Bytes

BITS & BYTES

File and Disk Recovery for Just About Anyone • You should have made a backup, but noooooo...	Steve Costa	57
A Picture of Everything a Computer Knows • A glimpse of assembly language for first timers	Scott Kronick	65
Using Font/DA Mover • Installing and removing fonts, DAs, and more	Stephen Howard	70
Editing Your DSAT • A ResEdit challenge	Christian Wiedmann	73
ResEdit: Hacking for the Rest of Us • Changing menus, cursors, dialogs, and icons	Christian Wiedmann	75
The Animator: A Simple Lightspeed Pascal Program • A programming exercise using animation	Linda Custer	78
ZBasic • A quick review by a BASIC user	Ken Huie	85
Bypassing the Trap Dispatcher for Fun and Profit • A hackers tool for speeding up applications	James E. Hopper	88



File and Disk Recovery for Just About Anyone A Beginner's Guide

Steve Costa

Well, it's late at night and you are just putting the finishing touches on your thesis, financial report, Great American Novel, etc. and a Mac owner's worst nightmare happens. The power goes out, or someone pulls your plug or lightning strikes and... when you reboot, you look at your Mac screen and see...



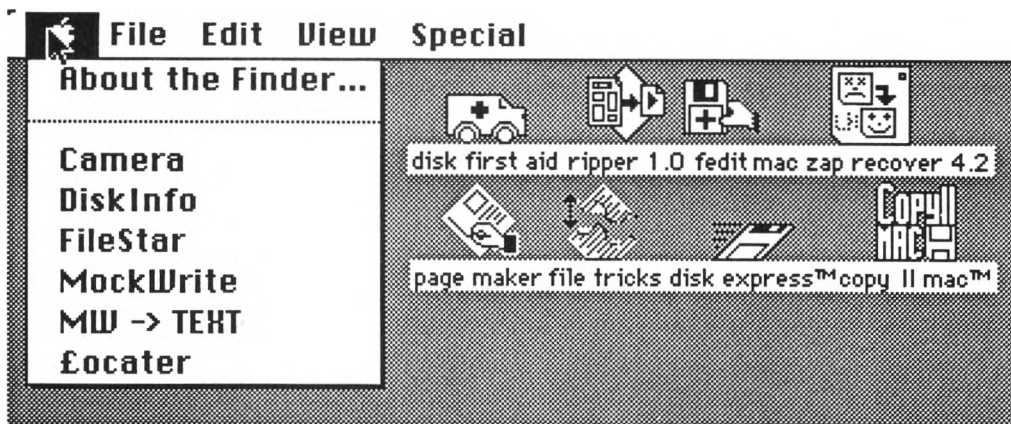
Do you panic? No, you cleverly boot up your Mac with another System and stick the offending disk in your external drive, but the gods this time weren't listening because you see in the middle of your screen one of the Macintosh's worst error messages...



Do you panic now? Nope! You're just beginning to fight!!!

Hmm, enough of this for now... How does one recover from this type of error? Here at BMUG, we receive one way or another about 5 to 10 floppies a week and a few hard disk drives that fit into this type of situation. And we are able to repair or recover about 90% of the disks and data. All it takes is the right software and a little knowhow.

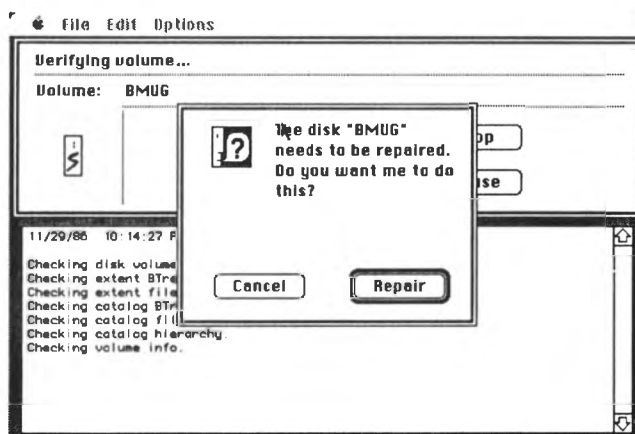
First, the software that you will need: I'll explain why and how to use the following programs to help recover, hopefully, your lost files and disks.



Some of the programs are ShareWare, like *DiskInfo* and *File Star* and others, like *Mac Zap Recover* are commercial products.

To get started, you need to make a copy of your floppy disk with *Copy II Mac*, or any other sector copier. Do not just drag from the Finder. Do not make a bit copy. Bit copiers make too good a copy of your disk. A sector copier may actually fix your disk by copying new repaired sectors while a bit copy copies over the damaged sectors as they are.

Now is the time to determine whether this will be easy or time-consuming. The easy way out is by using *Disk First Aid*, a program from Apple that comes with all new Apple Hard Disk drives. *Disk First Aid*, is aptly named. When it works, it fixes and repairs and gives you back your disk in good shape. It does have a few drawbacks as it only works with HFS volumes. So, if after running *Disk First Aid*, you get this message:



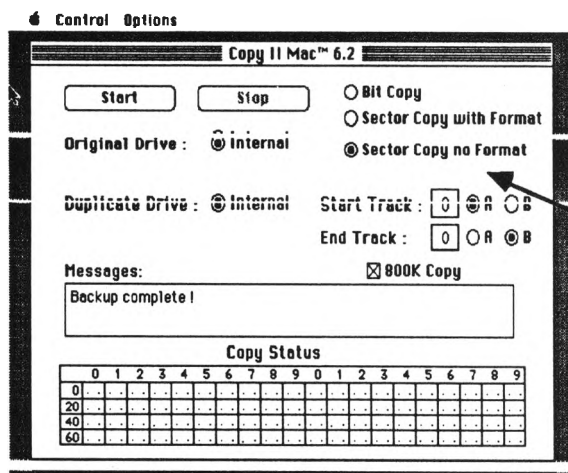
You do want to click on the **Repair** button. I've never known it to make a mistake and leave a disk worse than it was before it started. If it tells you that this disk "Cannot be Verified" it can mean any number of things, some of them worse than others. Still, *Disk First Aid* may be able to help. *Disk First Aid* can do something that no other file recovery programs do or do well. It



can mount your floppy or hard disk even if the directory is corrupted., which is the major cause of all Macintosh disk errors. The BTree and catalog hierarchy are all part of the Macintosh directory. (You can see *Disk First Aid* checking the directory in Fig. 1) Once you have the disk mounted, you can use *HFS Locator* or *FileStar* to copy your files off the damaged disk or drive. If this fails with a floppy disk, there is still hope. Use the **Search** command of either *File Star* or *HFS Locator* and after you find the file you want, go to **Copy** and copy it to another disk. With *Disk First Aid*, and assorted DA's, you can recover about 30 to 40% of bad disks and drives. This is all that can be done at first to recover and repair whole disks as opposed to files on a disk.

Now it gets harder. To go on, we need one more thing: BMUG Disk #50. BMUG Disk #50 is a special 800K MFS (Macintosh File System) disk that has allocation blocks set at 1 while a normal 800K MFS disk has them set at 4 and an HFS (Hierarchical File System) disk has them set at 8. What this means is that we can play around with a damaged 800K HFS disk and Fedit 1.08 and recover most of the files on the disk. BMUG Disk #50 has a special track 0, which is where the various directories are kept. It is a completely blank directory; it has no Desktop file or any of the other nice things which make a Mac disk a Mac disk.

After making an 800K copy of your damaged disk, set *Copy II Mac* to copy only **track 0**, and copy BMUG Disk #50's track 0 over to your copy of your damaged disk. Be sure that "**Sector Copy no format**" is selected.



Be sure this this is selected!

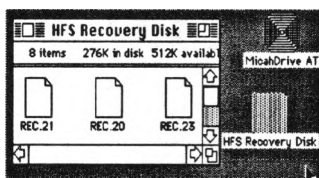
After copying over track 0, go to the Finder. Your damaged disk should be mountable from the Desktop now.



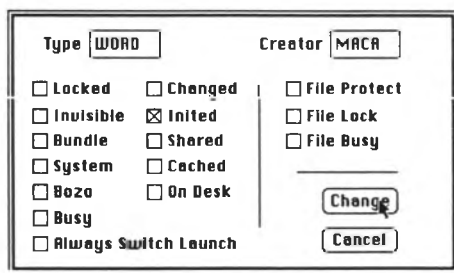
As you can see, it's completely blank, and an MFS 800K disk besides. These days it's still easier to recover MFS disks than HFS disks. Now you need to go into Fedit 1.08.



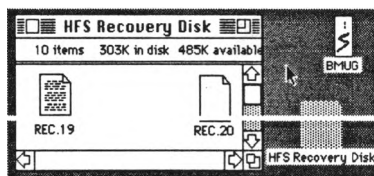
Once in Fedit, go to the File menu and select Volume to open your damaged disk. Now off to the Special menu and down to "Recover Deleted Files". Because we replaced the damaged directory on track 0 with a blank track 0 from BMUG disk #50, we can recover files on the disk as if they had been thrown away.



While Fedit can recover these files you still can't open them yet. As you can see, all you get are files with the icon that the Mac gives files that it doesn't know what to do with. But we can change that by setting the Type and Creator. To set the Type and Creator you need to go to the Display menu and select "Finder Attributes". the box that pops up will have 2 small buttons to fill. Put whatever the correct Type and Creator are.



In MacWrite's case the correct Type is WORD and the Creator is MACA. As you can see, it works.



Now we are through with the easy part. If none of things that I've just gone over have worked we need to go to a few of the other programs on my Disk Fixer disk.

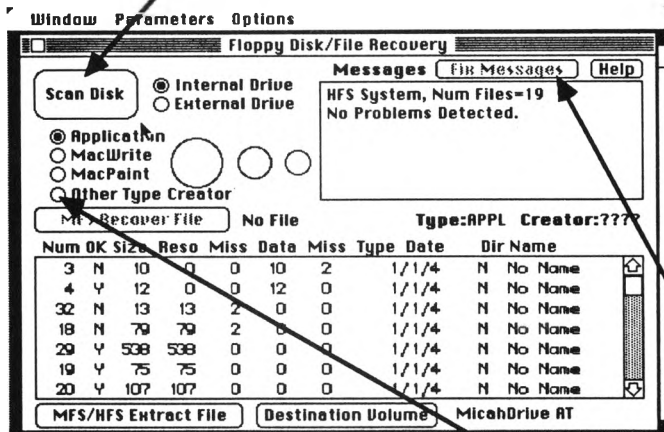


Next we need to take a look at *Mac Zap Recover 4.2*. *Mac Zap* can do a few things that *Fedit* can't do such as recover files that are damaged in some way. It will do its best to recover most of a file if it can't recover all of the file due some data that has been damaged.

First, launch *MacZap*.

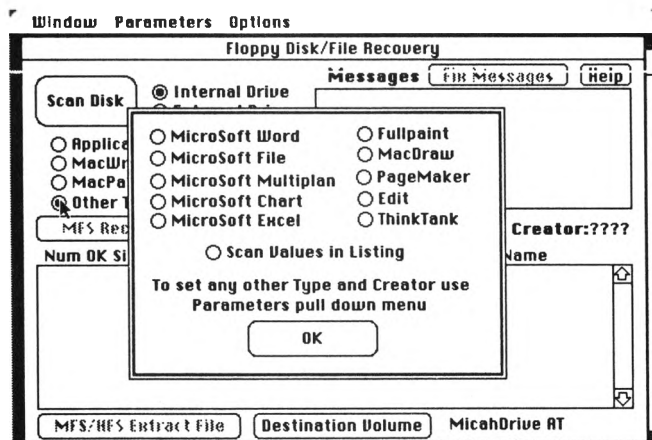


Now go to Scan Disk. Mac Zap will scan the disk and tell you what files it thinks it can recover. Mac Zap 4.2 is still not completely HFS compatible as it cannot tell you what the names are of any files on an HFS disk. So it can take some detective work



to recover the file that you want. You can compare the size or what kind of File it is. To extract the file go to "MFS Recover File" if it is a 400K, and "MFS/HFS" Extract File" if is an 800K disk. If Mac Zap thinks it can repair your disk, the Fix Messages button will be highlighted.

It sometimes works but only do it after it extracts all the files it can. Mac Zap can automatically set the Type and Creator by setting the defaults in the Parameter menu.



Or you can set them from "Other Type Creator". Every file that you extrac will have the Type and Creator that you've set.

Applications like *MacWrite* or *PageMaker* are mostly resources and most data files are mostly data with a resource or two. *PageMaker* files are the exception. They are almost completely resources. There still are a few ways to recover files from *MacWrite* which I'll go over.

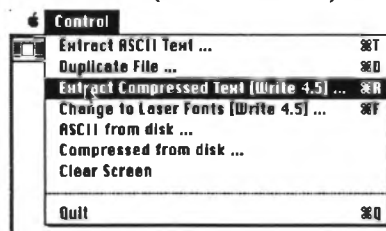
There are two ways that one can go, depending on what kind of file you have to recover. In our disk of tricks there are a few programs, such as *PageMaker* and *Ripper*, that are good for only one thing. *PageMaker* is here because it can open up many corrupted *MacWrite* files. The most common cause for a *MacWrite* file is a corrupted "end of file" segment. *MacWrite* and *Word* need to know how long a file is before they can open it. *PageMaker* is one of the few programs that doesn't need to know this information to open a *MacWrite* file.

1. To start, you need to load *PageMaker* and *MacWrite* into *Switcher 5.01* (or greater). Then use the **Place** command to load the *MacWrite* file, which should slide right in with all the correct formatting. One page at a time, put it in the clipboard and switch to *MacWrite*. (Be sure and select **Always Convert Clipboard** in *Switcher* options.)

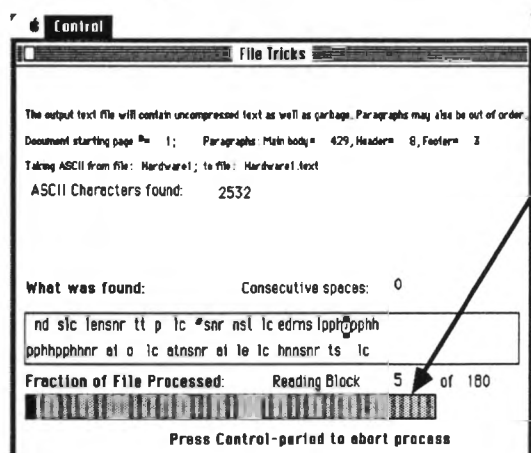
2. *HabaWord* is another program that doesn't need to know where a *MacWrite* file ends. Use the **Paste Document** in *HabaWord*. Try *Word* as well. If the *MacWrite* file is too corrupted, none of these tricks will work. Then you have to use another program called *File Trickss*, a very handy program.



To use *File Tricks* to recover *MacWrite* file, or even a few other things, go to the Control menu and select "Extract Compressed File (MacWrite 4.5)". What that will give you is a



text file. You will lose all your formatting but you will have something to work with. The other selections like "ASCII from Disk" and "Compress from Disk" are to be used in order of desperation.

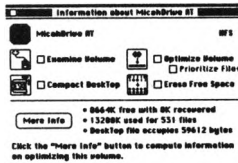


File Tricks will uncompress your *MacWrite* file and save it as text file and show you what it is saving and show you how far along it is.



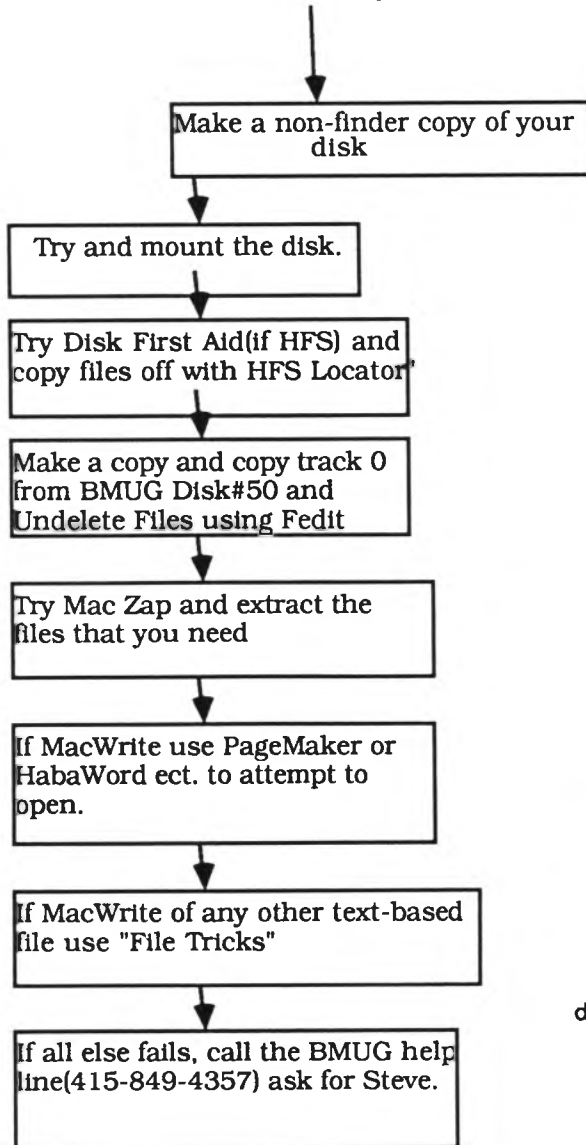
Ripper is a great program for recovering *PageMaker* files. It will take a *PageMaker* file and save it as an Acta file with the graphics and text separate. A *PageMaker* file is almost all resources. If one of its many resources is corrupted *PageMaker* will not open it up. *Ripper* will though! But if the resource map is corrupted *Ripper* will not be able to help.

Last but not least is *Disk Express* which sometimes will repair minor directory problems, though that isn't what it's normally used for.



Disk Express, by resetting your allocation blocks to what they should be, can solve a few of a damaged disk's problems. And it's a good utility for speeding up a drive that has been used for a long time without being reformatted.

What to do if you have a bad disk.



copy II mac™

Copy II Mac is a product of Central Point Software
9700 SW Capitol Hwy.
Suite 100
Portland, OR 97210 \$29



mac zap recover 4.2

Mac Zap \$49
Micro Analyst, Inc
PO Box 15003
Austin, TX 78761
MacMasters \$40
108 E.Fremont Ave.
Sunnyvale, CA 94087



fedit

ShareWare \$10
James E. Hopper
7135 Springbaro Pike
Dayton, OH 45449



ripper 1.0

Joseph F. Buchanam
University Of Utah



file tricks

Apple Dealers
Bill Bruffy
Karl B. Young



disk first aid

All Whipple \$29
Larry Davis
ALSOFT
PO Box 927
Spring, TX 77383



disk express™1.06

HFS Locator

PBI Software
March Gosnell

\$15

FileStar

ShareWare
Searle Software Innational
Mukilteo, WA 98275

This article is just a primer for beginners in file recovery. BMUG Disk #50 will consist of this article plus additions as time goes by. It will be a two-disk set with the various PD (public domain) and ShareWare file recovery tools. If people have any questions please write or call and I'll do my best to help.

Steve Costa is on his way out the door to pick up a quesadilla. after a long day of recovering files for yet another frantic computer user.

MacLabeler by Harry Critchfield

If you ever had thoughts of :

A-I would like a label that stands out in a certain way

B-If only I had a label with a graphic on it then I could...

C-There must be a better way to keep my disks from getting mixed...

If any ideas like these, have come to mind, then MacLabeler ought to be tried. It is a way to solve any of assorted problems that one has with identifying certain software files in a quick manner. The MacLabeler ad tells what can be done with the software in a very concise manner (usually a 1/6 page ad - toward the back of a Mac magazine). The blank labels that are included with the software work extremely well, when used for printing your newly designed labels. MacLabeler costs \$ 49.95.



A Picture of Everything a Computer Knows

Copyright ©1986 Scott Kronick (all rights reserved)

This is an excerpt from *Kronick's MPW Book*, a 300+ page introduction and tutorial to the Macintosh Programming Workshop with emphasis on beginning assembly programming. If my publisher doesn't fork over big bucks soon, you can get a copy directly by writing me at: 1442A Walnut St. #278, Berkeley, CA 94709.

1.1 Slot is My Name, Hex is My Game

Get ready. You are going to see a picture that represents everything a computer knows. By the time you understand each element of this picture, you'll understand how a computer works. Other chapters in the MPW Book describe the elements of the picture in detail. In this article, you are asked only to recognize the primary element: hexadecimal numbers.

Assembly language programmers use hexadecimal numbers constantly. Hexadecimal is a counting system based on sixteen different digits. The digits look like this:

0 1 2 3 4 5 6 7 8 9 A B C D E F

The first ten digits are the same as those used in the decimal numbering system. The numeric quantities ten, eleven, twelve, thirteen, fourteen, and fifteen—which require two digits to be expressed in decimal form—are represented by A, B, C, D, E, and F in hexadecimal form. In program code, hexadecimal (hex) numbers are preceded by a dollar sign (\$) and decimal numbers are preceded by a pound sign (#).

For example, \$A is equivalent to the decimal number ten (10). \$F is equivalent to fifteen (15). \$4 is equivalent to four (4). Later, you will see how hex numbers larger than \$F (decimal 15) are expressed.

1.2 Different Slots for those Delicate Spots

Next to this book, a debugger is your finest tool for learning about assembly programming. However, it is inevitable that your experimentation with debugging will cause programs and disks to crash, bomb, seize, freeze, or otherwise fail in a way such that your only recourse is to reset the Macintosh. Protect yourself from catastrophe by always having backup copies of all your work disks.

The following illustration is a snapshot of the Macintosh as it manipulates hexadecimal numbers. All numbers are hex unless otherwise specified. This snapshot, and the others to follow, can be reproduced by using the MacsBug debugger (as described in another chapter of the *Kronick's MPW Book*).

```
401F52:                                SUBQ.L  #$2,A7
PC=00401F52      SR=00002004      TM=0000084A
D0=00000000      D1=00000002      D2=00000000      D3=00000000
D4=00000000      D5=00000000      D6=00000000      D7=00000000
A0=00F80000      A1=000EF9F2      A2=00000000      A3=000EFC1A
A4=0000FF5E      A5=000EFBF8      A6=000EE5A0      A7=000EF9EC
>
```



The snapshot displays a very small part of the Macintosh's memory. Yet this small part is particularly important because of the 6-digit number shown on the top line along the left column. The number is the current value of the *program counter* (PC). Immediately below the 6-digit number, you can see the information repeated: PC=00401F52

A Macintosh functions by manipulating single *slots* of information. The 68000 processor, the heart of the machine, can keep track of only one slot at any one moment. The PC identifies the slot that will currently be acted on by the processor.

Everything a computer knows is stored in slots. The PC itself is a slot. It so happens that the PC slot contains information about the slot currently being acted upon by the processor.

The two most important revelations about slots are:

1. All the information that a computer knows is contained in slots.
2. When slots refer to other slots, they do so by using addresses or, in a few special cases, names.

The 68000 processor uses eighteen special slots, called *registers*, that are identified by two-character names. Ordinary slots have no names; they will be referenced by hex numbers representing *addresses*. Register slots serve as convenient, speedily-accessed storage, while ordinary slots are the primary components of the Macintosh's memory.

The names of the registers are:

PC, SR, D0, D1, D2, D3, D4, D5, D6, D7, A1, A2, A3, A4, A5, A6, A7

Last section's snapshot shows that the contents of these special slots are hex numbers. Computer information is stored numerically; the most common numeric form used in assembly language programming is hexadecimal.

The fear and loathing section below introduces the word "bit," a common programming term that designates the smallest unit of computer information. Slots are filled with bits of information. That's all the technical gruel for now.

1.3 Up Bit Creek (A Fear and Loathing Sidetrack)

Machines are very intelligent. Field spaniels are fairly intelligent. Woodchucks have sharp teeth, but routinely err when solving binary arithmetic problems. How intelligent are you?

Before there were intelligent people, people could count only to two. Starting with zero, they would say, "0, 1, uh, let's go hunting for a bit to eat." Ever since, a bit has come to represent the value 0 or 1.

The counting of fingers and toes had not been discovered yet. People had two hands, two feet, two eyes, two ears. Men had two of some things that women did not and women had two of some things that men did not. Nature provided men and women with an instinct to put one important thing together, but only after their intelligence evolved could they put two and two together.



Machines are very intelligent, but nature did not bless them with the fun things given to men and women. As a result, machines lead boring lives. They can put two and two together till hell serves Häagen Daz. Yet they always sleep alone at night. Would you like to be a machine?

Study the following illustration titled "Nude Ascending a Staircase".

```
401F52:                                SUBQ.L  #$2,A7
PC=00401F52    SR=00002004    TM=0000084A
D0=00000000    D1=00000002    D2=00000000    D3=00000000
D4=00000000    D5=00000000    D6=00000000    D7=00000000
A0=00F80000    A1=000EF9F2    A2=00000000    A3=000EFC1A
A4=0000FF5E    A5=000EFBF8    A6=000EE5A0    A7=000EF9EC
>
```

Machines appreciate the fine aesthetics of abstract art. People like to look at nudes. Are you contemplating the subtle nuances of hexadecimal notation as the artist intended, or is your gutter-mind composing an image of flesh in motion?

At this point in the article there ought to be an illustration of a girl and a boy playing volleyball on the beach. Most machines would regard such an illustration as an uninteresting pattern of black and white pixels, however its inclusion here would reflect the economic observation that machines don't buy books.

There comes a time in the life of every girl and boy when special feelings—glorious feelings nurtured in the innocence of youth—blossom into a powerful yearning to abandon the traditional counting system and experiment with base sixteen numbers. The numbers 10, 11, 12, 13, 14, and 15 become insufficient to express the majesty of ten, eleven, twelve, thirteen, fourteen, and fifteen. They are replaced by A, B, C, D, E, and F, respectively. A dollar sign (\$) before these hexadecimal digits distinguishes them from ordinary decimal numbers.

"When was your first sexual experience," the young woman asked, serving a soft underhand across the net.

"I was \$E years-old. A friend of my big sister said she wanted to show me her stamp collection." He set the ball up, then tapped an easy shot near to her. "When we got to her room she said she wished I was a stamp and started licking me. How about you?"

They volleyed noncompetitively as she answered. "I was \$12. We were in love. I wasn't the kind of girl who would go to base sixteen on the first date. We finally did it the night before he was being shipped out to an accountancy training program with Arthur Anderson and Co. It was wonderful. I said goodbye dreaming of our life together. But a few weeks later, I got a letter from his family. There had been an accident. It was horrible. A top shelf had collapsed. A torrent of 3-ring binders filled with Chevron's fiscal '86 subsidiary ledgers crashed down. They said the poor boy never knew what hit him.

* * *

In the January 1987 issue of *High Tekkie Times*, the journal of semiconductor paraphernalia for those under the influence of recreational substances, a critique of "Nude Ascending a

previous work delved into the deevolutionary work of his great grandfather, Franz. The younger Kafka had provided credible evidence that through sustained psychopharmacological technique, one could indeed turn into a giant lizard.

Writes Kafka: "In 'Nude Ascending a Staircase', we have the classic lines of a cubist's desultory self-esteem confronting the neo-impressionism of a post-modern romanticist the day after he has come down with a rash. Of course, on an entirely different level, we have a snapshot of a two-bit computer program splayed open with its disassembled guts assaulting the visual sense of all god-fearing souls.

"Of one thing we can be certain: the artist has intense contempt for humanity. The work reflects misogyny at its most despicable—a vulgar rendition of anatomical features without the slightest hint of feminine warmth and compassion. The probing digits line up like columns of soldiers programmed for mayhem. The lewd abbreviations suggest acts that exceed binary decency. The portrayal of characters locked into garish equations offends nature's symmetry, demeaning both subject and medium under the flimsy guise of abstraction.

"That voyeurism has penetrated the microprocessor arena should come as no surprise. Every since the microscope began revealing squirmy flagella and single-celled ova, science has made a point of unearthing disgusting art forms. Imagination has been picked bare by algorithmic vultures. No tourniquet can arrest this kind of pestilence. If the fire must be fought—and I dare say it must—we shall attack with an artist's fervor, with gossip and innuendo to make electronic canvasses such as 'Nude Ascending a Staircase' the laughing stock of every chic gallery, boutique and eatery in the land."

The next month *High Tekkie Times* published this letter in response:

Editor,

Kafka missed the point in his critique of "Nude Ascending a Staircase."

The artist never intended the work to be viewed as an end product. Rather, it is a stepping stone in the pursuit of pleasure from the heart of a young machine. Like a rainbow that leads to a pot of gold, the work paints the horizon with resplendent hues of code to lead the intrepid adventurer through shared memories.

For instance, look at the item titled PC. (Kafka calls them equations, which ought to tell you something about the man whose prior efforts in the field of pharmacology chronicled his Tuesday newsstand visits for the purpose of inhaling the ink vapors off fresh copies of *Modern Bride* magazine.) Here, memory is displayed starting at the current value of the program counter (PC).

While the memory of a machine might seem as convoluted as the table talk of two trial lawyers discussing stir-fried goat intestines, certain tools make memory observation easier. Notice the 5-digit number on the top line. This portrays the "throb of the heart" captured in time. With each throb, a new memory image springs forth. The characters below may be gibberish to you, but to a machine it might carry the memory of its first day out of the carton, admired, caressed by warm hands, gently stripped of packing materials, and, in the consummative act, plugged into a grounded outlet.

Some say PC stands for Program Counter. Others believe it is short for Passion



Consummated. In either case, the artist has drawn a picture that illustrates, across a single line, the current throb of a machine's heart and the specific memory associated with that throb.

This is just an example of the artful magnificence that Kafka failed to recognize in his wordy morass. Works such as these show the heart pouring hexadecimal lifeblood to a young machine. If, only for a moment, Kafka would lift his drug-addled head, he would discover a refreshing perspective above the rim of his commode consciousness.

Mr. Moss

1.4 See Dick and Jane Grimace: First Lines of Assembly Code

This section shows you a single line of assembly language code. This is an example of a line of code that a programmer types onto the keyboard to create an assembly language program.

The example instruction looks like this:

```
MOVE.L D1, (A1) +
```

Since everything that a computer knows is contained in slots, programs are written to manipulate the slots. Pascal, C, and other programming languages manipulate slots implicitly. Assembly language, from which other languages are created, manipulates slots explicitly.

D1 and A1 are the names of special register slots. MOVE.L is one instruction from a set of instructions available to manipulate the 68000 processor and its memory slots. The example line of code also uses a space, a comma, parentheses, and a plus sign. Your task is to understand how all of these components work together to manipulate information in slots.

If you are guessing that this example of assembly language code was chosen as an introduction because it is among the simpler, you are wrong. It is among the more difficult. As such, you are not expected at this point to understand what the code does. The purpose of this section is to reaffirm two facets of programming that you already know:

1. Slots (with register names for those inside the processor and hex addresses for those that make up memory) hold all the information a computer knows.
2. Assembly language moves, rearranges, tests, adds, subtracts, multiplies, divides, inverts, shifts, branches to, compares, and otherwise manipulates slots.

Msg. #1411 in **Public Messages** Posted on 06/02/86 at 23:36:27

To: ALL From: MICHAEL BROWN

Subject: Grateful Dead at the Greek

This may be wholly inappropriate for this space, but here goes. I would really like to see the Dead at the Greek this year but was unable to get tickets. If there is anyone out there who might have some extra, I would love to go. leave a message for me here and I shall get it that day.
Thanks!!

Using *Font/DA Mover*

Stephen Howard

What is it?

Font/DA Mover is a utility (i.e. an application that does something terrifically useful but not very clever) which allows you to control the fonts and desk accessories (DAs) available to you when using your other applications. It's a necessity for anyone using a Mac because of the hundreds of fonts and desk accessories to choose from. Most Mac owners know how to use *Font/DA Mover*. If you know one of these knowledgeable persons, go and have him or her show you how to use it – it's far easier to learn by watching than by reading about it – but for the record and for those Mac users on the frontiers of civilization, here goes.

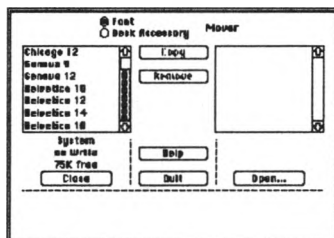
Points to Remember

Fonts and desk accessories are resources that exist in your System file (resources : System file :: peas : pod). *Font/DA Mover* allows you to move these resources in and out of your System, or if you prefer, to store them in separate storage files (*Font/DA Mover* = Pea Mover; storage files = canned peas).

System files can hold a lot of fonts and DAs, as can the storage files, but each size of font and each desk accessory (autonomous pea) takes up memory. So, there is a practical limit to the number of each you can hold on a disk. Hard disks and 800k floppies somewhat alleviate this problem. There is, however, a very real limit to the number of DAs that can be installed in a System file: 16 (that's it, maximundo). Remember, the fonts and DAs you put in a System file are the ones available for use when running under that System, the ones in storage files are inaccessible until installed (So you may have shelves and shelves of canned peas, but you cannot do anything with them until you take them out of the cans and put them in pea pods – which, of course, is something no one ever really does with peas, but it is something you might do with System resources).

How to Do it

Put *Font/DA Mover* on a disk with a System file (your Systems Tools disk comes perfectly prepared). Start *Font/DA Mover* (Clever Trick Reference #1 – look at the list at the end of this article to learn short-cuts and tricks). The screen pops up, and looks like this:



Fonts appear in the left box, listed individually by family name and size. The right box is blank. Beneath the left box is the name of the file you are looking at (in this case a System) and the amount of free memory left on the disk. By clicking the **Open** button beneath the right box, you summon up a dialog box with which you can select another System file, an existing storage file, or create a new storage file (Clever Trick References #2&3). Once the desired file is chosen and opened, its contents will appear in the right box (if it has no fonts in it yet, the box will remain blank, but the name of the file will still appear beneath the box). To move a font from the left box to right, click on the one you want to move (Clever Trick Reference #4), and



then click the **Copy** button. Nothing will happen for a while and then the font you selected will appear in the right box.

That's it. You have just copied a font resource from a System file to some other file. If the other file was also a System file (perhaps on a different disk) then you have successfully installed a the font as an available resource. Congratulations.

Gosh, That Seems Simple

It is. And the same process applies exactly to moving DAs. In fact, the process is also reversible - you can move stuff from right to left the same way. You switch from moving fonts to desk accessories by clicking the little round radio button at the top (the one labeled DA).

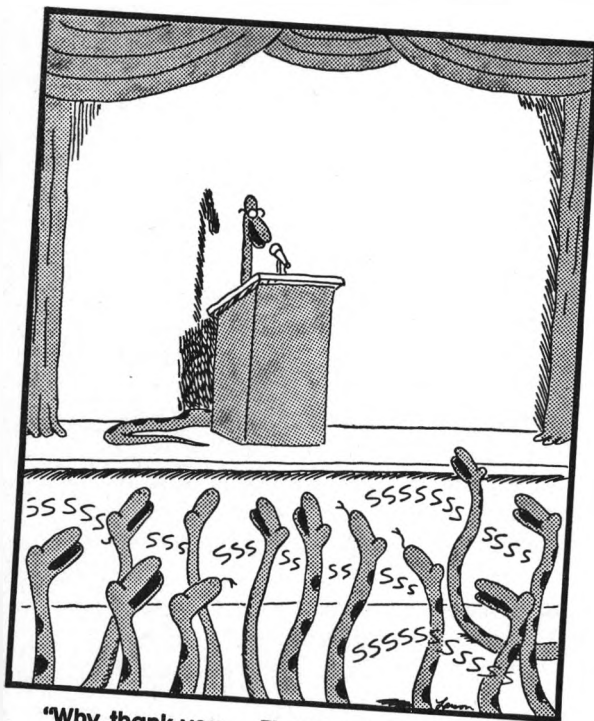
...than one font or DA at time by dragging across a bunch of them or by clicking **Copy** will operate on the whole by the fonts or DAs you have selected. Compare this to the free disk memory its are going to fit where you want them.

more dangerous, and it is. If you select a be erased (*Font/DA Mover* = pea eater). e, you simply have no use whatsoever for could then use *Font/DA Mover* to remove ree up some memory. Before you do that, le into a storage file in case you miss it

re any of the four System fonts (Monaco 9, The Mac needs these fonts to run its dialog e a dialog box will pop up telling you that Control Panel desk accessory from your ay crash if it's not around.

re installing the screen versions of these magewriter and Imagewriter II, but not the 'lus.

ing *Font/DA Mover*, it will display desk



"Why, thank you. ... Thank you very much!"

2. By holding down the Option key when clicking **Open**, you can get a listing of *all* files on the disk. You can, therefore, install fonts or desk accessories directly into an application or even into a document so that these resources will be available - in addition to the fonts and DAs in the Sytem - when using that application or document. Using this trick allows you to install more than 16 DAs: you put 16 in the System file and then up to 16 more directly into an application for a grand total of 32!
3. Holding down the Option key when you click on the **Close** button beneath the display box ejects the disk on which that file is stored.



4. If you wait a second after selecting a single font size, an example of that font will appear at the bottom of the *Font/DA Mover* window.
5. If you hold down the Option key when selecting a font or DA, the amounts of memory in the data and resource forks for that selection will appear at the bottom of the *Font/DA Mover* window.

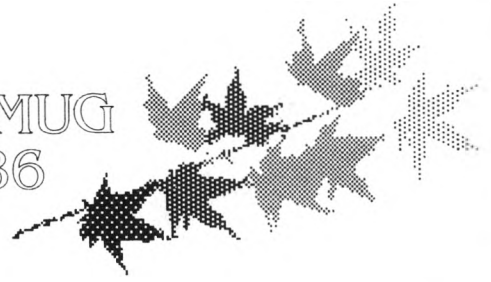
Msg. #1827 in **Public Messages** Posted on 06/26/86 at 16:59:11
To: ALL From: JIM BARCLAY
Subject: External Monitor
Does anyone know if you can interface a Mac with an external Monitor?
If so, how is this done?

Msg. #1840 in **Public Messages** Posted on 06/27/86 at 08:48:00
To: JIM BARCLAY From: SYSTEM OPERATOR
Subject: Reply To 'External Monitor'
Jim -

There are a couple of ways:

- a) BMUG uses a board from PDS Video Technology, in San Diego. It puts out HI-RES video, so you need a special monitor to cope with it.
- b) A similar board is available from Mentauris, location unknown.
- c) You could get the CHROMATRON from Beck-Tech: only \$6000 (!), it puts out pure NTSC & RGB video, and, as a side effect, gives you color!

-- Raines Cohen
SYSOP, BMUG BBS



Editing your DSAT

Christian Wiedmann

A recent BMUG newsletter article described a way to change the "Welcome to Macintosh" message to any other message, 20 characters or less. This article describes how to use ResEdit to replace "Welcome to Macintosh" with a message of any length, and also how to replace the icon in the startup dialog box. Note that this process is not too easy to do.

The System Startup dialog box is stored in a resource of type 'DSAT', not the normal 'DLOG' dialog box resource type. All 'DSAT' definitions contain a resource ID followed by the size of the rest of the definition. To change just the icon and the text of the 'DSAT', all we need ineed to modify are the definitions of the icon and text. An icon definition is simply the ID, the length of the rest of the definition, a rectangle where the icon will appear, and then the icon in bitmap form. The text definition is an ID, a length, a location (two words or 8 hexadecimal digits), the actual text, and a "00" suffix.

The first step to changing the string is opening up a new 'STR' resource. Type your string. Next, open the resource using the general template so you can see the hexadecimal bytes. Copy these. Now, find the string in the DSAT, and paste your string in. Add a "00" to the end of the string and remove the first byte of the string. Add five to that first byte, and replace the length byte with this number. Note that the "00" must be present. Also make sure the text lines up to an even position:

```
ss ss ss ss 00 xx xx xx
```

is wrong, where the 00 is the end of the text, xx is the beginning of the next definition, and ss is part of the string. Add another "00" at the end and update the length.

Example. The "Welcome to Macintosh." text definition looks like this:

```

                                00 32
00 1A 00 6C    00 C0 57 65
6C 63 6F 6D    65 20 74 6F
20 4D 61 63    69 6E 74 6F
73 68 2E 00

```

The outlined text is the ID, the length, and the point where the text appears. The underlined text is the actual message "Welcome to Macintosh." Note that 1A is *not* the length you would get as the first byte of a string when you type in the string. Also note the null ("00") character at the end of the string. You would select the shadowed text and replace it with your string and then update the "1A" to the correct length.

The icon is a lot easier to edit. Simply open a new 'ICON' and draw whatever you want. Open it as general and copy the hexadecimal bytes of the 'ICON'. Find the icon in the 'DSAT' and paste your icon's representation over it.

Here are the relevant parts of the default DSAT in Hex representation:

00 13 00 28	00 0A 00 32	The first entry is the startup alert table.
00 00 00 1F	00 00 00 00	It says primary text is "0032" and the
00 2A 00 0A	00 00 00 00	icon is "001F"
00 1E 00 00	00 00 00 2B	
00 0A 00 34	00 00 00 00	



BMUG FALL 1986

```
00 00 00 3C      FF F6 00 0A
00 35 00 00      00 00 00 00
00 00 FF F5      00 0A 00 36
00 00 00 00      00 00 00 00
FF F4 00 0A      00 37 00 00
00 00 00 00      00 00 00 32
00 1A 00 6C      00 C0 57 65
6C 63 6F 6D      65 20 74 6F
20 4D 61 63      69 6E 74 6F
73 68 2E 00      00 34 00 32
00 6C 00 82      57 61 72 6E
69 6E 67 20      2D 2D 20 74
68 69 73 20      73 74 61 72
74 2D 75 70      20 64 69 73
6B 20 69 73      20 6E 6F 74
20 75 73 61      62 6C 65 2E
00 00 00 35      00 18 00 7E
00 C0 4D 61      63 73 42 75
67 20 69 6E      73 74 61 6C
6C 65 64 2E      00 00 00 36
00 1C 00 92      00 C0 44 69
73 61 73 73      65 6D 62 6C
65 72 20 69      6E 73 74 61
6C 6C 65 64      2E 00 00 37
00 06 00 A6      00 C0 20 00
00 1F 00 88      00 59 00 38
00 79 00 58      00 07 FF 80
00 08 00 00      00 08 7E 20
00 08 01 20      00 08 01 20
04 08 01 20      0C 08 01 20
08 08 01 20      66 08 01 20
99 08 01 20      81 08 FE 20
80 08 00 20      80 08 00 20
4A 00 1F 20      34 00 00 20
00 FC 00 20      03 00 00 00
04 01 FF F0      04 00 00 08
03 E0 1F E4      00 10 05 52
00 10 00 A9      00 20 00 01
00 20 01 FE      00 18 00 00
00 04 00 00      00 00 C0 00
00 01 20 00      00 02 50 00
00 00 88 00      00 00 08 00
00 00 10 00      00 3C 00 1A
00 02 00 46      00 96 00 32
00 AA 00 64
```

Ah ha! Here's "0032." This must be the text
The length is 1A, the point "006C, 00C0."

This is a bunch of random information.
We don't care about it.

Here's "001F", the icon.
These two bytes are the actual start of
the icon definition. Start selecting here.

Select until the boldfaced part. Then paste in your icon definition. Note that this isn't the end of the 'DSAT'; a significant amount of data comes after this.

Christian Wiedmann is a part time MacHacker and a student at UC Berkeley. He is a prospective Computer Science major.



ResEdit: Hacking for the Rest of Us

Christian Wiedmann

ResEdit, also called the Resource Editor, is actually a developer's tool. It was designed to ease the development of windows, icons, dialogs, and many other nifty "tools" for the programmer.

Resources can be almost anything associated with an application. This includes the program itself, which is contained in 'CODE' resources. The easiest resources to modify, however are the things you see on the screen: icons, windows, fonts, menus, and dialog boxes are examples of these. With the Resource Editor, all these can be modified at will. Note that the things stored in the scrapbook are resources. Thus anything that can be in the scrapbook is a resource.

If you want to learn to use ResEdit, begin by booting a disk with ResEdit on it. (ResEdit is available on several BMUG disks.) The first screen ResEdit presents you with contains several windows, each containing a list of files of one of the disks your Mac currently knows about.

Editing a resource

To open a resource type, double click on its name. You are then given a list of all the resources of the selected type present in the file you're editing. To open the resource, select it, then choose open from the file menu or hit return, or simply double click on it. If ResEdit recognizes the icon type, you are given some kind of representation of the resource. If ResEdit doesn't recognize the type, it just displays a list of the hexadecimal bytes of the resource. When the resource is open, you can edit whatever representation you are given by ResEdit.

Example: Changing the trashcan icon

First select the name "Finder" by clicking on it. Select "open" in the file menu and you are given a list of four letter codes. Find 'ICN#' (for Icon List) and open up the resource. You are now given a new window which contains all the Icons in the finder. Double click on the trashcan, and you are given a window which looks a lot like FatBits in MacPaint. Clicking on any of the dots will cause the Icon to be changed. Notice that 'ICN#'s have two images: an actual image and a "mask". The mask defines which parts of the icon are opaque. The background shows through the places in the icon where the mask is white. The representations of the icon along the bottom of the screen show how the icon and its mask interact with various background patterns. When you are finished editing your icon, simply close the Finder window, and click **Yes** to save changes. If you accidentally mess up at any stage, choose **Revert** from the File menu to get back the original.

Menus ('MENU')

The main thing you can do to menus is add keyboard equivalents. This means when you hit the command key and the letter specified, the menu item is automatically selected. This greatly increases the speed you can use some programs, since often it is a waste of time to remove your hands off the keyboard to move the mouse around. Open the menu resource and find the entry which says "keyboard equivalent". Simply type the character you want there. Note that you can use Option or even Shift-Option when typing the character. Be sure not to duplicate keyboard equivalents as this sometimes can cause unpredictable results.

Cursors ('CURS')

Like icon lists, cursors have a mask, but in addition the mask, cursors contain a "hot spot".



This determines where in the icon the click counts. Experiment by moving it around while editing a cursor. To try the cursor, select "try cursor" from the CURS menu which appeared when you opened the cursor.

Windows ('WIND')

When you open a window resource, you get a miniature version of the Mac screen. To move the window, click anywhere in it and drag it around. To resize it, click where the grow box would appear and drag. You can also type screen coordinates for the top left and bottom right coordinate to define the windows location and size. Just select "show as text" from the WIND menu.

Dialogs ('DLOG' and 'DITL')

An example of a dialog box is the "Do you want to save?" message you get if you choose quit from an application without saving first. You can edit both the location and size and the contents of such a window. Alter the dialog window just like normal windows. Double click on the dialog to edit the items. When you do this, ResEdit opens the 'DITL' (Dialog Item List). You can move around and modify all the items in the window. The DITL menu allows you to resize the window containing the dialog items so it is exactly the same size as the dialog window would be.

Function keys ('FKEY')

Another example is the 'FKEY' (Function Key) resources. Function keys are Shift-Command and a number. You will notice that FKEY 3 and FKEY 4 are present in your System file. These contain the code to dump the screen to disk or the printer. You can remove them if you want to save space on your disk and will never need to save a screen image. You can't easily edit 'FKEY's, but you can copy and paste them between files. Lots of 'FKEY' utilities, similar to miniature DA's, are available on BMUG disk 27. These can be pasted in your system.

Pattern ('PAT' and 'PAT#')

You can edit the patterns stored in the system file by opening these resources. If for some reason you want to save space in the system, you can also remove some of these. Add patterns to the list by selecting **New** from the File menu when you have opened a pattern.

Miscellaneous items

Many resources are too complicated to edit yourself, but can be cut and pasted between files. You can do this with the resources found in the "Modify MacDraw" file on BMUG disk 20. This file contains alternate menus and dialog boxes which make MacDraw easier to use. This way you can distribute changes you think particularly useful to your friends. You can copy pictures in and out of applications too. If you want to save space in MacPaint, you can remove the Help and Shortcuts screens. These are stored in 'PICT' resources.

First things to edit

I have always found the 'ICON's in the system file which appear when you do something wrong very boring, so I replaced them. Also on my list are: menus in the finder, the trash can icon, disk icons, the watch cursor, and miscellaneous dialog boxes. In many applications, the default windows can be moved around to facilitate use of the program. In *MacPaint*, the paintbrushes and spraypaint patterns can be changed (see Fall 85 BMUG Newsletter). Play around with a copy of your favorite program and write an article describing the most useful changes you made to it for the next newsletter.



Here is a brief list of the resource types easily editable:

- ALRT- Contains an alert template.
- CURS- Contains a cursor.
- DITL- Contains a list of items contained in a dialog box. Each of these is associated with a DLOG.
- DLOG- Contains the definition of a dialog box.
- DSAT- Contains the system startup dialog box and other resources. This is not easy to edit. (see other article)
- FONT- Contains a font. It is probably easier to use one of the commercially available products to edit these.
- ICON- Contains an Icon used in dialog boxes, menus, etc.
- ICN#- Contains an Icon used by the Finder. (an icon and a mask)
- SICN- Contains a small Icon (a la Finder)
- MENU- Defines a Menu
- PAT - Contains a single pattern.
- PAT#- Contains a list of patterns.
- WIND- Defines a Window



Big Eyes/Callaway

The Animator:

A Simple Lightspeed Pascal Program

Copyright ©1986 Linda Custer

Lightspeed Pascal also makes your job easy. *LSP* is currently the only Pascal compiler for the Mac that supports source-level debugging. (That means—in human terms—that while compiling or running the program, when *LSP* finds an error, it will bring up your actual Pascal program and point to the line that caused the error.) *LSP* compiles very quickly (hence its name), and generates fairly compact code. As if that weren't enough, it also allows you to step

Animat

```

program Animator;
{ ; The Animator
{ ; rev 9 November 1986

{ ; Source code and application by Linda Custer
{ ; Certain parts of compiled code © 1986 THINK

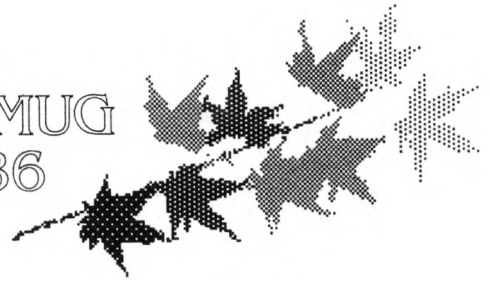
{ ; This application takes a set of 'PICT' resources and displays them in order—one after
{ ; the other. It supports desk accessories and has the ability to display the pictures in
{ ; reverse order as well as forwards. It is intended to show how a very simple Mac appli-
{ ; cation is built.

{ ; Thanks to Raines Cohen for the beta testing, and for the Backwards, Disk Insert,
{ ; CountResources, and autowindow suggestions.
{ ; Thanks for Andrew Singer for the Lightspeed environment.

uses
    UInit, USetup, UMainLoop;

```

Options	File (by build order)	Size
	MacPasLib	15842
	MacTraps	5440
	ROM85Lib	908
☐ N V R	ROM85	0
☐ N V R	UGlobals	0
☐ N V R	UEvent	934
☐ N V R	UMainLoop	394
☐ N V R	USetup	908
☐ N V R	UInit	38
☐ N V R	Animator	28



through and trace your program, set stops, observe variables on the fly, and support text and drawing windows automatically. These types of tools are usually only available to programmers working with a slow interpreter. They are rare in a compiler. *LSP* also supports the concept of a Project, so you get SmartLinking and compact code automatically without having to consciously manage various files.

If you are familiar with the *MacPascal* interpreter (also by THINK), you will feel at home with *LSP*. Most *MacPascal* programs will compile immediately in *LSP* with the addition of only two commands: *ShowText* and *ShowDrawing*. If you have been using *MacPascal* up to this point, you will probably want to switch to *LSP* to gain compiled speed while your programs are running.

Nothing is perfect, so of course *LSP* has some flaws. If you are writing programs for yourself or are not overly concerned with the size of your application, *LSP* will suit you fine. It will shorten enormously the time it takes you to write a bug-free program. However, if you are writing a program to sell commercially, or if disk space is precious, you will probably want to develop your program in *LSP* and then recompile a working version with either *TML Pascal* or Apple's *Macintosh Programmers' Workshop Pascal*. Compiling the finished code with another compiler may save you a few K in the finished product and may make your program run just a little faster. In general, *LSP's* rules are a subset of the rules used by other compilers, so moving code *out* of *LSP* should be painless. Moving code *in* to *LSP* might take a little more work.

If you are developing a very long application, you may run up against *LSP's* limitations on code length. *LSP* encourages you to write your code in *units*, and no unit can be longer than about 2000 lines. Since an application can have a very large number of units, this requirement should not keep you from writing a program of the complexity of, say, *Excel* or *Omnis3*, but you may have to juggle some code around occasionally. If you can be disciplined enough to write all your code in modular units, you'll find that recompiling will be much faster than if your whole program is in just a few units. *LSP* is smart enough to only recompile what has been changed.

The Animator

If you need a little help to get started programming, take a look at a small program I put together in *LSP*. The source code and the compiled program for this article are available on the BMUG Fall 1986 membership disk, so you can take a look at the necessary resources using *ResEdit* and you won't have to type in the code yourself. (*Important:* Without the appropriate resource file, this program will crash and burn when you compile and try to run it!) I call it *The Animator*, and its sole reason for existing is simply to take (almost) any number of previously-stored 'PICT' resources and animate them in a window that can be dragged but not closed or resized. It doesn't do much, but it illustrates the concepts of a main event loop and drawing in windows. It also illustrates how to store strings, menus, and pictures as resources and load them from your program. *Animator* supports DAs (as any good application should—you'll see that it's not hard), and gives menu options for quitting and reversing the animation order. You can use the *Animator* source code as a shell, cutting a little of the code, and adding a lot of code to make a program that does whatever you like.

The structure of *Animator* is shown in in Figure 1. Normally, with a program this small, you would be able to include all the necessary code in just one file. I have separated this program into six files, however, to illustrate how units would be used for more complex programs.

Unit by Unit

As soon as you begin to examine any *LSP* source code, you'll note that it follows the same structure that most modern Pascal compilers do: that of **units** made up of **interface** and **implementation** sections. Whenever you want to include a **unit**, you say a **program** or a **unit** **uses** another **unit**. The idea of **units** originated with UCSD Pascal, and has gained widespread acceptance. So with *LSP*, all projects that generate stand-alone applications have only one **program** but an almost unlimited number of **units**. Some projects like desk accessories and 'FKEY's may be made up entirely of units only.

The first unit in *Animator* is not a unit at all—it is the main program. It simply initializes a few things and falls into the main event loop. It has no “clean up” routine because we don't create any files that we need to save for next time or modify any system globals. The next unit, *UGlobals*, has **const**, **type**, and **var** declarations for identifiers used throughout the program. As with any good Pascal program, *Animator* uses numbers sparingly throughout the code so that parameters can be changed easily by changing their global definitions.

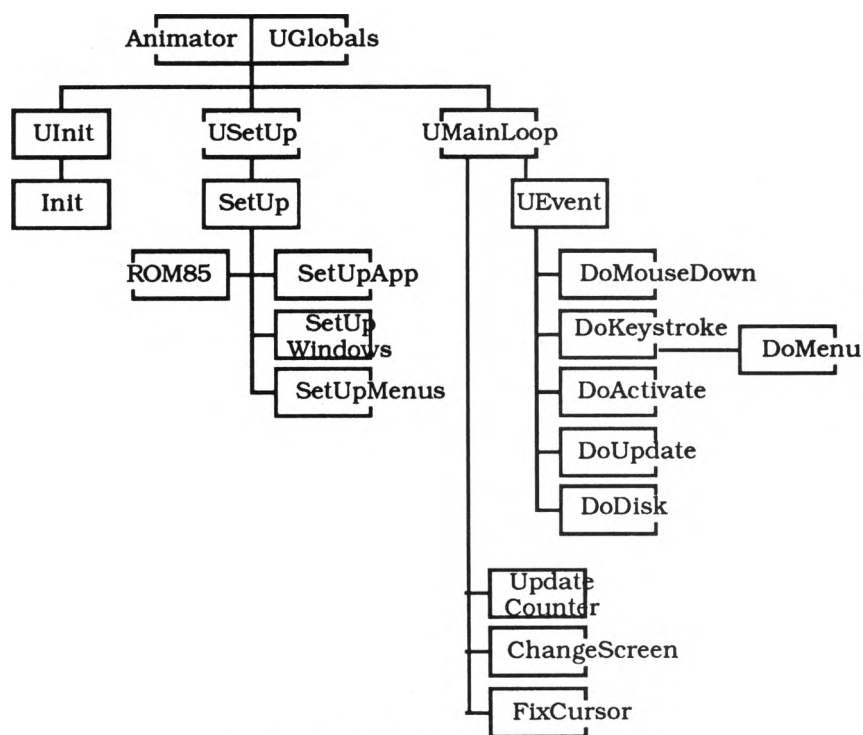


Figure 1. Schematic of Animator

The next unit, *Uinit*, notes as a comment that *LSP* automatically performs the Macintosh equivalent of a “witch chant” at the beginning of every program (unless you explicitly tell it not to). The “witch chant” initializes the Toolbox. If you compile this program with *TML* or another compiler that doesn't do Toolbox initialization automatically, you should make the bracketed comments real Pascal procedure calls. *Uinit* then calls **MoreMasters** four times. Since the Macintosh's memory management scheme uses master pointers to point to almost everything, and since the “things” can be relocated at any time, their master pointers are the only things that *don't* move. By calling **MoreMasters** at the very beginning of our program, rather than waiting until the program needs more master pointers later, we put all the master pointers that we will need (and some extras for desk accessories to use) at the very bottom of our



heap. We thus avoid placing chunks of immovable master pointers in the middle of our application heap. Each call to **MoreMasters** allocates a block of 64 master pointers. Lastly, UInit calls **FlushEvent** to rezero the event queue so our application won't have to deal with any events that weren't processed by previous applications.

USetUp gets the menu and picture resources we need for the rest of the program, puts the menu bar (including all the system desk accessories) and application window up on the screen, defines how far the application window can be dragged, and initializes some global variables. It counts the number of 'PICT' resources contained in only the application file (with **Count1Resources**) and loads each 'PICT' in turn into a doubly-linked ring of pointers. After it loads the first 'PICT', it checks to see that the 'PICT' was really available. If it wasn't, **GetPicture** returns a **nil** handle, so we notify the user that no 'PICT's were available, then quit. USetUp also calculates where to place the window to center it on the screen and make it just a little bigger than the size of the stored 'PICT's. I believe the window placement code will center Animator's window on *any* screen, even a Radius or E-Machine screen, but I haven't tried it yet.

If you compile *Animator* to work on a MacPlus or 512K Enhanced Mac only, leave USetUp just the way you found it and include the ROM85 library and ROM85 files. If you compile it to work on *any* Mac, then you'll have to make a change to USetUpApp. I originally wrote *Animator* to contain only definitions found in Chernicoff and the old ROMs, but I didn't want the code to specifically compile to show only a fixed number of 'PICT's. The **CountResources** call seemed to do the trick at finding out how many 'PICT's were available. However, it included 'PICT's in the system file as well as those in the application file. Only the new ROM call **Count1Resources** fixed the problem. So, for an old-ROM Mac, leave out the ROM85Lib and ROM85 files and define MaxPict as a constant (equal to the number of pictures you want to display) rather than a variable. Eliminate the **Count1Resources** call.

The Main Event

After we've initialized everything, we're ready to enter the main loop, contained, curiously enough, in UMainLoop. This is the loop that sets every Macintosh application apart from the rest and waits for the user to tell the computer what to do. First, we call **System Task** to give desk accessories a chance to do things they need to do. Every application should call **SystemTask** once each tick (1/60th of a second). Although calling it once each time through the main event loop is usually sufficient, if your application is doing something that will take a long time inside of the main event loop, you should add **SystemTask** calls there as well. Next, we call **ChangeScreen** to invalidate the application window. Rather than doing the drawing within the main event loop, invalidating the window leads to an update event that will redraw the window later. Drawing the window immediately might be more expedient in a simple program like *Animator*, but generally drawing is better left completely to update events. Once more than one window is doing more than one thing, code can get tangled quickly and update events simplify things enormously. **ChangeScreen** also updates the handle to the picture to be drawn by traversing the linked list of available 'PICT's.

Now, we're ready to check with the operating system to see if we have to do anything. A call to **GetNextEvent** returns a boolean value: TRUE if an event is waiting, and FALSE if no events are waiting. Since we invalidate the application window every time we go through the main event loop, we'd better take care of that update event first. Otherwise, we will always have events waiting in the queue that we'll never process. So, we call **GetNextEvent** with a mask that only fetches update events. We pass the message from **GetNextEvent** to our update procedure. Then we call **GetNextEvent** again, this time with a mask that fetches all events. If the result is true, we use a **case** statement to process the event. Lastly, we call **FlxCursor** to find out whether the



BMUG FALL 1986

mouse has been still for at least TickWait ticks. If the mouse has been still, we call **ObscureCursor** to hide it until the mouse moves again.

The screenshot shows a window titled "UEvent" containing Pascal code. Overlaid on this is a smaller window titled "LightsBug" which contains a list of resources and a memory dump.

UEvent Code:

```
begin { DoUpdate }  
  TargetWindow := WindowPtrToFront(resourceId);  
  if TargetWindow = nil then  
    begin  
      { ; Update events should be handled by the  
        BeginUpdate(TargetWindow);  
        DrawPicture(Picture, TargetWindow);  
        EndUpdate(TargetWindow);  
    end  
  end; { DoUpdate }  
  
procedure DoDisk;((MountDisk))  
{ ; Handle disk-insertion  
{ ; mounted properly  
  
var
```

LightsBug Resource List:

registers	zones	DoUpdate	MainLoop	Animator
H 010B1A	001867	2	PICT	0080 1D8
H 010B16	00185C	2	PICT	0081 1D8
H 010B12	001839	2	PICT	0082 1D8
H 010B0E	00182A	2	PICT	0083 1D8
H 010B0A	00001A	2	STR	0080 1D8
H 0106FA	0001A2	0		
H 010ADE	000048	2	MENU	0003 1D8
H 010ADA	000066	0		
H 010AE2	000031	2	MENU	0002 1D8
H 010AE6	000181	2	MENU	0001 1D8

Memory Dump:

Base: 000000 (+) Offset: 0000 (Edit)

000:	00F8	0000	FFFF	FFFF	000F	69F6	.0..0000..i0
00C:	0000	CFF0	0000	CFF2	0000	CFF4	..00..00..00
018:	0000	CFF6	0000	CFF8	0040	1134	..00..00..0.4
024:	000F	67D6	000F	75E8	0000	CFFA	..g+.u0..00

UEvent handles all the individual events. **DoMouseDown** asks where the mouse was clicked. If it was clicked in the menu, **DoMenu** finds out where and opens a desk accessory, displays the About Animator... alert, or sets the Backwards or Quitting flags. It also sends **Edit** menu commands to desk accessories for processing. If the mouse was clicked somewhere else, **DoMouseDown** appropriately selects or drags a window, passes the click onto a desk accessory, or ignores the click.

DoKeydown is simple in *Animator* because it only has to process menu equivalents chosen with the command key. The **MenuSelect** call processes command-key equivalents for us. Activate events are processed by **DoActivate**. Besides just bringing selected windows to the front of the screen and highlighting them, **DoActivate** enables and disables the **Edit** menu so it is unavailable to the application but is available for desk accessories. **DoUpdate** is the routine that refreshes the application window on every pass through the main event loop and really does the drawing. Lastly, **DoDisk** takes care of mounting disks and allowing the user to initialize unformatted disks.

Left as an Exercise for the Reader

You can customize *Animator* by creating any 'PICT' resources (using MacPaint, MacDraw, etc., and copying the images into the Scrapbook) and pasting those 'PICT's into *Animator* to replace the ones already there. Start numbering the 'PICT's with resource IDs of 128 and up (sequentially). Each 'PICT' should be the same size, but you can add as many 'PICT's as you can fit. Each 'PICT', from FirstPict (currently 128) up will be displayed in turn.

BMUG Fall 1986



Animator is fairly robust. I'm sure I've violated a rule here and there and overlooked a few things, but *Animator* checks for a few simple errors. If a desk accessory does not load (maybe because enough memory is not available), *Animator* displays an alert to let the user know. If no 'PICT's are available, *Animator* just stops and quits. If the available 'PICT' resources lead to a window that is bigger than available screen space, the window is made as large as possible and only some of the 'PICT' is shown. Despite these checks, I have undoubtedly left some stones unturned. I have compiled *Animator* with an 8K stack and a 128K heap, and loading in lots of large 'PICT's will certainly lead to an out-of-memory error. As an exercise to see if you're awake, you might want to add a few lines of memory checking to preview how many pictures are to be drawn, how large each is, and how much space is available. If you find that not enough space exists, display the (as yet unused) ID=204 alert and **ExitToShell**. Or you might want to get fancy and ask the user if they mind displaying just as many 'PICT's as will fit, or maybe try growing the heap zone to accomodate the 'PICT's. The possibilities are almost limitless.

Linda Custer is a Ph.D. candidate in Berkeley's Chemical Engineering department and a freelance Mac writer, programmer, and consultant.

Msg. #3948 in **Public Messages** Posted on 10/20/86 at 22:00:37
To: ALL From: GEORGE LANGDON
Subject: Pascals for the Mac
I concur with Claude that LightSpeed Pascal is a good chioce for both learning Pascal on the Mac and obtaining a useful, compiled language. I have been at it for about three weeks. LS is really the same as MacPascal from Apple...actually both where written by Think. The syntax is the same, but with LS you can write real programs. The price is right as well. I paid \$125 at Winners Circle, have seen it for \$100 at Software for Less, and in MacTutor there is an ad listing a price of \$77. After reading through Moll and a couple for other standard Pascal books, I got frustrated with their limited info on programming for the Mac. I'm now working through Christopher Morgan's book 'Hidden Powers of the Macintosh' from Plume. Well written and actually easy to understand if you work through the programs in the book. Most can be entered directly in LS Pascal, but beware...some have outright bugs that should have been caught in editing. Hopes this helps and responds to a couple of message at once. GO FOR IT!!

BMUG
FALL 1986

Msg. #3144 in **Public Messages** Posted on 09/11/86 at 00:08:31

To: ALL From: VANCE BRADFORD

Subject: Lightspeed Pascal

I recently purchased Think Technology Pascal, and wow!!!! It has a fantastic symbolic debugger, just like the original Macintosh Pascal, and it's really, really fast! It also has the best user's guide and reference manual I've seen for a compiled development language. I am really impressed! Think Technology has done it again.

One mans opinion,

Vance Bradford, Murray, Ut.

Msg. #3335 in **Public Messages** Posted on 09/18/86 at 20:13:05

To: ALL From: GEORGE LANGDON

Subject: LightSpeed Pascal

Well, I may be the first on my block to get a copy of Lightspeed Pascal. I happened upon a copy at Winners Circle weekend before last. After two weeks I can only say that it appears quite solid, very easy to use, and basically a compiler version of Macintosh Pascal. I am no Pascal programmer, and so have been using Hayden's book "Intro to Macintosh Pascal" to learn. The programs and instructions for using the language are identical to LightSpeed. LS does have a very slick source code debugger; however, I haven't gotten into any jams to use it, yet. I have found that longer programs cannot be entered as one source file. LS has a limit of 32,000K on file size; so you must break-up longer programs into Units. Not really difficult, just another thing to learn. The LS manual does not teach you much about Pascal. Anyone have good suggestions on primers on Macintosh Pascal??

If you have questions, just ask...

Msg. #3483 in **Public Messages** Posted on 09/26/86 at 15:34:09

To: ERIK TIEMROTH From: RAINES COHEN

Subject: Reply To 'pascal compilers'

Erik -

I like LightSpeed Pascal a lot, as it is blindingly fast and a terrific debugging environment. It has a couple of bugs but they are usually work-around-able. I saw Turbo Pascal for the Mac yesterday at the PC Faire in San Francisco, and it looked VERY fast but it doesn't offer the kind of debugging that LightSpeed does. TML, with version 2.0, supports objects (like MacApp and MPW Pasccal), and now also runs faster.

-- Raines Cohen

Co-founder, BMUG Beginning Developers SIG

(meets every other Wednesday in Wheeler hall) (call 849-9114)



ZBasic: A Quick Review by a BASIC User

Ken Huie

Like the ad says

ZBasic is a BASIC compiler that is able to produce stand-alone applications and maintain the interactive (quick and dirty) nature of BASIC. It also includes many of the extensions of Microsoft (MS) *Basic* useful for programming the Mac interface, such as the MENU, ON MOUSE, etc. commands, although they are implemented somewhat differently. The disk is sold with the *ZBasic* application (100K), a help file (45K), *Edit* (61K), *RMaker* (11K), and the standard System/Finder.

Outwardly, the *ZBasic* program is clean and well organized; however, I was unable to use it without first reading (the first Mac test) the manual, a 500+ page softbound book. *ZBasic* has a "configure" selection on the menu bar to change the type of default variable (integer, long integer, single precision, double precision) and set the number of BCD digits. Variables default to integer form and I was forced to start using type declarations (\$,!,#) at the end of variables' names (goodbye quick and dirty). By the way, calculations with real numbers using BCD math are slower than integers but numbers with up to 240 digits (with no round-off) can be controlled. Cryptographers will love this. Labels are supported but they must be enclosed in quote marks, one of several differences from MS BASIC.

Noticeably missing is the ability to declare arrays using a variable. All arrays must be declared with an explicit number. [ie: DIM AnArray(25) vs. DIM AnArray(AVariable)] Although *ZBASIC* advertises portability between the Macintosh, IBM PC, Z-80, and Apple II versions of *ZBASIC*, this portability is more theoretical than real for those wanting to implement the Mac interface, or utilize features such as serial port or MacinTalk access.

The editors

ZBasic comes with *three* editors. Two editors operate within *ZBasic* and one is the familiar application *Edit*, which has a lot of features, fonts and the ability to show invisible characters.

ZBasic's command window contains the second text editor: a traditional line editor. It operates quite like a scrolling ASCII terminal with a command structure similar to *MBASIC* on a CP/M machine. This is expected since *ZBasic* is available on Z-80, Apple II, and MS-DOS machines. Listings within this window have auto-indenting of loops. The **Edit** and **List** commands are the only way to view text in this window. When you run a program without specifying an output window, the output appears in this default window.

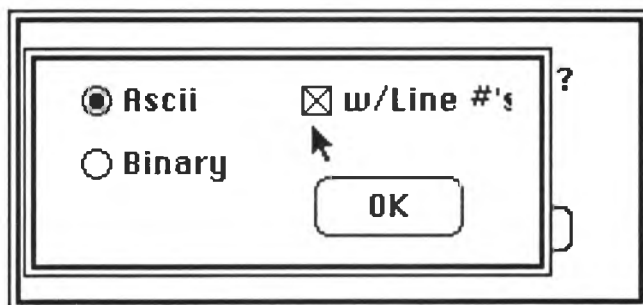
The third editor, toggled in/out with Command-E, is a Mac-type text editor that allows mouse movement and scroll elevators. This window editor is sometimes faster than the line editor but you cannot RUN from this window; you have to toggle back to the command window using Command-E. In early versions of *ZBASIC* this editor barely worked at all; as of version 3.01 it still exhibits idiosyncracies such as the occasional misplacement of lines.

When entering lines inside the command window, *ZBasic* needs a unique line number on each new line of text. The auto-indent feature is not present in the text editor window. Although *ZBasic* has an auto-line-number function and existing text files (like from *Microsoft BASIC*) receive line numbers when you import them, you can only remove line numbers from existing files by utilizing an undocumented feature.



BMUG FALL 1986

To save without line numbers, click on the close box of the command window after making a change to your program. ZBASIC will then respond with the dialog box "Save changes before closing?" Click on the **YES** button. This will bring up the following dialog box:



Click off the **w/Line #s** button, and **OK**, and you're home free.

No simple "search and replace" function is available within the two ZBasic editors. This is annoying, since ZBasic is case sensitive to variable names and finding a variable name with an incorrect uppercase letter can prove to be frustrating. A **Find** command, provided as a substitute for "search and replace", is very awkward to use. One easy way to get around this is to install a text editor desk accessory such as MockWrite that implements **Find**.

Documentation

The book has around 500+ pages, with about 100 pages devoted to MSDOS, CP/M, and Apple II specific commands and considerations. The section on Mac specific commands is about 115 pages. About 20 pages discuss how to convert from other BASICs, and includes a convenient list of commands *not* available in ZBasic and what command to use instead. The remaining 400 pages cover commands and graphics common to all versions. Lots of multi-line examples are scattered about. The book has a 6 page index that helps find most needed topics. MacInTalk, AppleTalk, and serial port access are present as special functions and the toolbox is minimally supported, but you need *Inside Macintosh* to sort out the procedure calls. The book simply warns that you "use them at your own risk" (ugly).

Mac Interface Support

The easiest way to program the Mac interface in ZBASIC is to utilize the built in commands for creating windows, implementing menu choices, scroll bars, help windows, and edit fields. Alert windows, BREAK, DIALOG, MENU, MOUSE, and TIMER event trapping, BUTTON support, FILES\$ (which brings up the familiar file saving or selection dialog boxes), and laserwriter support are all implemented, although you'll find yourself having to write example programs to figure out exactly how they work. Serial port and MacInTalk access is provided via the OPEN statement. To get serial port access (or statements such as INKEY\$) to work correctly, remember to turn off the BREAK, DIALOG, MENU, MOUSE, and TIMER interrupts.

ZBASIC supports in-line inclusion and execution of machine language code, via the CALL statement. Users programming the Mac interface in ZBASIC will outgrow the documentation within a short period of time. One possible source of information is the ZBASIC section within the Macintosh developers group on GENIE.

Some results

Typing "RUN" error-checks and compiles the program. Since the Sieve program benchmark is published in the ads for ZBasic, I chose to try some programs already written in Microsoft



BASIC not so much to see how fast they run but to test the compile time and ease of conversion to a stand alone application.

The first program "Hello" in *ZBasic* is equivalent to this familiar program in Pascal.

<pre>10 PRINT "Hello" 20 END</pre>	<pre>PROGRAM (output); BEGIN WRITELN('Hello') END.</pre>
------------------------------------	--

The second program is from a recent Byte article (Aug 86), "Palindrome", that uses long precision BCD math and lots of *STRING\$*. This program *RUNs* like a jet-set bandit.

<u>Program</u>	<u>Compile time</u>	<u>source size (bytes)</u>	<u>code size (bytes)</u>
"Hello"	39 sec.	19	34342
"Palindrome"	35 sec.	1713	36720

Note that these times are the times needed to compile to a 400K disk and possibly varies with the most recent disk operation; 800K disks are a bit faster. *RUNing* a program without compilation to disk is a more practical way to use *ZBasic*. Compiling destroys the ability to interact and debug a program. Test any program a lot before you bother to compile it. The size of the code generated while building the application demonstrates the lack of an intelligent linker; the whole library is added to each application. An equivalent program (Hello) in Pascal compiles to an application of less than 4K.

Conversion from Microsoft *BASIC* to *ZBasic* can be tedious and is non-trivial; although an MS *Basic* to *ZBASIC* translator is available, it does not handle all aspects of the conversion. For example, *ZBASIC* implements event trapping differently, requires quotes around statement labels, uses numbers rather than labels in a *RESTORE* statement, and has different syntax for *IF-THEN-ELSE* constructs. The default variable type has to be selected and the type declaration has to be added to all variables not of the default type. Upper and lower case spellings have to be matched. Re-numbering of lines is automatic but groups of lines are not easily moved; manual re-numbering is needed—a royal pain in a large program. The *Edit* application helps some, but the only test is to *RUN* the program and wait for crashes (some fatal, with bomb). In fact, since the programs are compiled when *RUN*, some *simple* errors are fatal if error trapping is not turned on. Turning on the *BREAK* function—so you can halt the program with command-period—slows the operation of a program noticeably, and will disrupt serial port access and keyboard reading via the *INKEY\$* function.

Value

At about \$90 (around \$50 for mailorder) *ZBasic* is a moderate value. The documentation is readable and accurate. The applications produced have a minimum size of about 34K. The saving graces of *ZBasic* are its speed (on the order of available C compilers), the interactive nature of *BASIC*, and the ability to produce stand alone applications that implement the Mac interface without learning the Toolbox calls. Version 3.01 does not fully support HFS, but version 3.02 should. Learn another language if you don't already know *BASIC*.

ZBasic, \$89.95
Zedcor, Inc
4500 E. Speedway, # 93
Tucson, Arizona 85712

Bypassing The Trap Dispatcher For Fun And Profit

James E. Hopper

Car 54 Where Are You?

One of the secrets of the Apple II success story was the support provided to the hackers. Apple actually published the source code for the ROM's in the reference manual (the one that replaced the early Red Book). With this wonder of documentation, a programmer could dramatically cut down the work in writing an application by calling the ROM routines (kind of familiar isn't it). However like most blessings it was not unmixed, and Apple is still paying for documenting these entry points as physical addresses in the ROM. This has forced them to maintain each of these entry points at the same address with each revision. Thus if a ROM procedure gets longer, there is usually no alternative but to put the end of the procedure off somewhere in the ROM and have the beginning portion jump to it. This has tended to turn the Apple II ROMs into a maze of patches, and made it very difficult for the ROM to be upgraded.

With the Macintosh, Apple had a second chance. What they did was to have the ROM install a dispatcher in RAM at boot time, and all access to the ROM is intended to be through this procedure. Thus it is possible to patch or change the address of a procedure in ROM by simply changing the address in the RAM resident Trap Dispatch Table. When the ROMs come out, the procedure addresses in ROM are moved about. Thus, if everyone follows the guidelines in writing their applications, they will all work without having to change the software with every new ROM release. The price paid for this is that every call to a ROM procedure has some overhead added to it as the dispatch procedure is called whenever a TRAP is executed. Thus, an extra call is made for every ROM command encountered. While normally this overhead is very small, if a trap is called enough times, it can add up and as a result slow down the execution of the program. This is especially true for those procedures whose execution time is comparable to the time to call them through the Trap Dispatcher.

Where Is That Bypass?

Some programmers have been known to speed up their code by calling ROM procedures directly, bypassing the trap dispatch procedure. This can be a very bad practice if done incorrectly, that is, by hard coding the ROM address into the code. While the resulting program will work, the first time the ROM is changed the program may crash and burn! One should note that it doesn't take a ROM revision to change a procedure's address, the 'INIT' and 'PTCH' resources both change the Trap Dispatch Table addresses for ROM procedures every time you boot your system. This is done to patch bugs or add features to existing ROM procedures. However if your program is calling the procedure in the ROM it is likely bypassing these fixes. A correct way to bypass the trap dispatch procedure is suggested in Scott Knaster's book, and is just as fast as the wrong way. Call the trap **GetTrapAddress** once in your program and get the current address of the trap procedure you are interested in. As Scott says, Apple will guarantee that the ROM procedure's address will not change while you are running your program.

Things Can Get A Bit Sticky!

To use this technique in a higher level language, like Pascal, you need what is called a "glue routine". What this Glue routine will do is correctly set up the arguments and call the ROM procedure at the address you got from the **GetTrapAddress** call. For example, in TML Pascal the call to the ROM procedure **BitTst** is defined as:

```
Test := BitTst(bytePtr, bitNum);
```



The compiler will place the proper arguments (bytePtr,bitNum) on the stack, then insert the Hex \$A85D in the object code. The use of the **inline** directive obviates the need for a glue routine in this case. The code generated by the compiler follows:

```
;;      Test:= BitTst (@bytePtr,1);
        clr.b          -(SP)
        lea            bytePtr(A5),A4
        move.l         A4,-(SP)
        move.l         #1,-(SP)
        dc.w           $A85D
        move.b         (SP)+,D6
        move.b         D6,test(A5)
```

Space is reserved on the stack for the Functions result of **BitTst**. A pointer to the first byte in the block is pushed onto the stack. A "1" (for first byte in the block) is pushed on the stack. It should be noted that using the actual numbers in a program is frowned upon. However, this is a hack, not a real program. The 2 bytes of the trap address are then put into the code (that's what **inline** really means). This will cause the processor to trap (or intercept) the instruction and jump to the trap dispatch procedure. The dispatcher returns control to the next statement after the trap is executed. This statement pops the function result off the stack, then moves it into the Pascal variable Test. This scheme must be slightly modified to bypass the Trap Dispatch Table, and will require the use of a Glue routine. Our Glue routine, which will replace the standard **INLINE**, will need the address of the trap procedure as an additional argument. It must clean up the stack to insure the arguments are correct. Then and only then will it do a jump to the ROM procedure. The call is as follows:

```
Test := NTBitTst(bytePtr,bitNum, BitTstTrapPtr)
```

The code generated by the compiler is as follows:

```
;;      Test := NTBitTst(bytePtr,1, BitTstTrapPtr);
        clr.b          -(SP)
        lea            bytePtr(A5),A4
        move.l         A4,-(SP)
        move.l         #1,-(SP)
        move.l         bittsttrapptr(A5),-(SP)
        jsr.w          NTbittst
        move.b         (SP)+,D6
        move.b         D6,test(A5)
```

The differences here are small, but significant. First, one more argument is pushed on the stack, BitTstTrapPtr. Please note that at this point, the stack is identical to the stack for the dispatcher based call, except for this last argument on the stack. The next instruction is a JSR to our glue routine, **NTBitTst**, instead of the **inline** trap address, which will push the return address onto the stack. The end of this routine is the same as before.

Listing 1 is the assembler source for a **BitTst** glue procedure which bypasses the Trap Dispatch Table. The Pascal compiler code copies the three arguments to the stack, as we saw above. The return address is then pushed on the stack and control is transferred to the glue procedure. The glue procedure then pulls the trap address off the stack, pops the return address off the stack,



BMUG FALL 1986

and puts it back on the stack where the trap address was (which is where the trap procedure expects it). While the stack manipulation looks a little strange, if you read your processor guide you will find that it is the most efficient way (most people when they see this for the first time tend to believe the compiler screwed up).

Listing 2 is a small program that I wrote to time how long the **BitTst** trap takes, calling it via the Dispatch procedure verses calling it directly. The sole purpose is to find out what the time difference is between the two methods of calling **BitTst**. Since all the code is identical except the calls to **BitTst**, the delta time is the time saved by bypassing the Trap Dispatcher.

Cruising 675!

When this test procedure is run, the time on my Mac+ for the dispatcher version is 54 ticks, while the time for the non dispatcher version was 38 ticks, or approximately 70% of the dispatch version time, note that the absolute times here are quite small. Thus this technique, as I mentioned at the start of this article, is useful only if you are going to call the trap a lot of times. What you might ask would call a trap procedure this many times? Well, for instance, a recent program John Matthews and I wrote converts an Apple II HiRes screen into a Mac bit map. This procedure calls **BitTst** 51,570 times for each picture, and **BitSet** for as many of these times as the Apple II HiRes bit is on.

This technique used for both **BitTst** and **BitSet** cuts slightly over 1 second out of the conversion time of 8 seconds using the dispatch table. While this is not a huge difference every little bit counts and only required about 10 lines of assembly code. Those lines can be added to my library of standard procedures for later reuse. Ok, I'll admit it makes little to no difference in this application but it was such a neat hack I couldn't resist. Besides it would have saved a larger amount of time on a full page graphic, or a 300 dpi bitmap for use with the Laserwriter. Let's see, an 8x10 bitmap at 300 dpi is 7,200,000 bits. If it called this routine that many times theoretically would save 192 seconds. Now that's more like it. I suspect this technique is only going to be of interest to those folks trying to squeeze the last little bit of performance out of their application, or perhaps image processing where you have a lot of bits to twiddle.

From Apple-Dayton Journal (Oct 1986), the newsletter of the Apple Dayton Users Group, Dayton, OH.

Jim Hopper is a long time Apple computer enthusiast, who bought his first Apple back in 1979. He has written articles for Call A.P.P.L.E. and Apple Orchard, in addition to the Apple Dayton Journal. He has written several PD utilities for both the Apple II and the Macintosh, including DiskScan, FastBoot, DiskMap, Sleep Key, DiskScan, and AppleIMac. He also has the distinction of writing the first arcade game for monkeys! He is does not however have his own personal picture icon like all the big name authors, yet (grin).

Listing 1

```
XDef NoTrapBitTst
;Function NtBitTst(BytePtr:Ptr;BitNumber:LongInt;TrapAddress:Ptr):Boolean;
    NtBitTst
; the stack has one too many arguments for this trap
; procedure so we must take the address off the stack and move
; the return address of our calling program in its place in order to
; insure the trap procedure sees the stack the same way it
; would if we had called the Trap Dispatcher
; get the address of the trap procedure from stack
```



```

        Move.L          4(SP),A0
; restore the stack to the way the procedure expects it
        Move.L          (SP)+,(SP)
; jump to the ROM procedure (which will return to the caller)
        JMP             (A0)

```

Listing 2

```

program TestDispatcherTime (Input, Output);
{$T APPL JIMH      }
{$I Tecmar:SoftwareDevelopment:PascalSystem:MemTypes.ipas  }
{$I Tecmar:SoftwareDevelopment:PascalSystem:QuickDraw.ipas }
{$I Tecmar:SoftwareDevelopment:PascalSystem:OSIntf.ipas    }
{$I Tecmar:SoftwareDevelopment:PascalSystem:ToolIntf.ipas  }
{$U Tecmar:SoftwareDevelopment:WorkingFiles:BitTst  }
{$a+}

var
    DummyAddress      : Integer;
    i                  : Integer;
    StartTime          : LongInt;
    EndTime            : LongInt;
    DispatchDeltaTime  : LongInt;
    NoDispatchDeltaTime : LongInt;
    BitTstTrapPtr      : Ptr;
    DispatchTimeString : Str255;
    NoDispatchTimeString : Str255;
    Test
    : Boolean;

function NTBitTst (BytePtr : Ptr; BitNumber : LongInt; TrapAddress : Ptr):
    Boolean; External;

begin
    { get the current address of the BitTst routine }

    BitTstTrapPtr := Pointer(GetTrapAddress($5D));
    StartTime := TickCount; { start the clock }

    { loop 10,000 times calling the function BitTst via the dispatch routine}
    { to get a fair sample }

    for i := 1 to 10000 do
        Test := BitTst(@DummyAddress, 1);

    EndTime := TickCount; { stop the clock }
    DispatchDeltaTime := EndTime - StartTime { compute the time }
    StartTime := TickCount; { restart the clock }

    { loop 10,000 times calling the function BitTst directly}
    { to get a fair sample }

    for i := 1 to 10000 do
        Test := NTBitTst(@DummyAddress, 1, BitTstTrapPtr);

```



BMUG FALL 1986

```
EndTime := TickCount; { stop the clock }
NoDispatchDeltaTime := EndTime - StartTime; { compute the time }

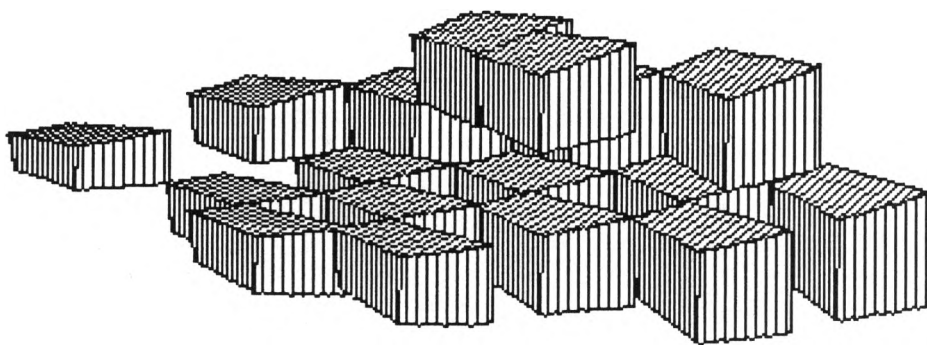
{ convert the times to strings for output }

NumToString(DispatchDeltaTime, DispatchTimeString);
NumToString(NoDispatchDeltaTime, NoDispatchTimeString);

{ print out the two times for the user }

Writeln('Dispatch time for 10000 iterations of BitTst: ',
    DispatchTimeString);
Writeln('No Dispatch time for 10000 iterations of BitTst: ';
    NoDispatchTimeString);
ReadLn; { let user read the results }

end.
```



Boxes/Hockabout

BMUG
Fall 1986



*Productivity
Tools*

PRODUCTIVITY TOOLS

File Finders	Mark Loncar	93
• Three programs that locate files		
Koala KAT-ing Around the Classroom	Kenneth William Hirsch	96
• A mouse alternative		
Micrograde and the Joys of Teaching Classes	Jasper Rine	98
• Grade bookkeeping and calculations		
All the Home Accounting Programs	Scott Beamer	101
• Five programs reviewed		
Mini Reviews	Scott Beamer	107
• On time, tax, and disk zapping		
Smart Alarms	Steve Cisler	110
• An appointment DA		



File Finders

Mark Loncar

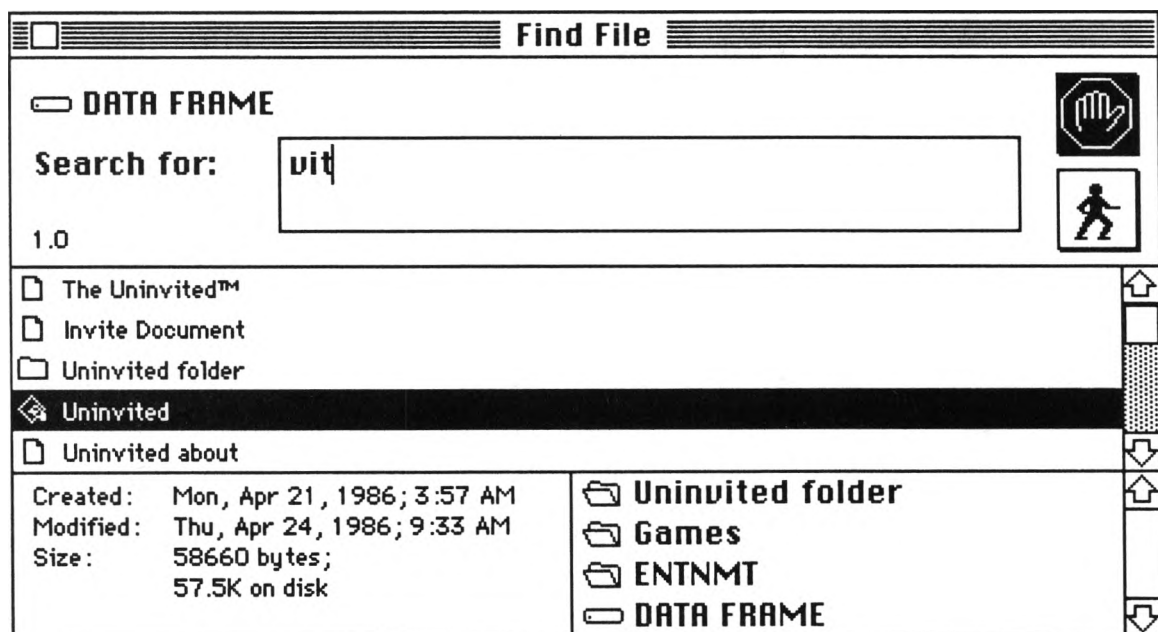
The Problem

As the prices of 20-megabyte hard disk drives drop, more and more Macintosh users are buying these mass storage devices to deal with their growing libraries of documents and applications. An unfortunate side effect of mass storage is the ease with which files that are hidden away in some forgotten folder can be misplaced or "lost" in the new Hierarchical File Structure (HFS).

Unlike the old Macintosh File Structure (MFS), HFS can only find files within the folder selected. If you have a document in a folder that is nested three or four generations deep in other folders, it may be hard to find, even if your system is generally well-organized. Luckily, there are Desk Accessories that can find a desired file, without having to leave the current application, by searching all folders for you. This article describes three of these filefinding DA's: Apple's *Find File*, Dave Dunham's *Disk Info* (v 1.43 or greater), and PBI Software's *HFS Locator Plus*. None costs more than \$40.00, and two are less expensive still.

Find File

Apple's *Find File* is the newest and least expensive. Released with Apple's new 20SC SCSI hard disk drive in September, 1986, and installed with the Font/DA mover, it takes up 11.5k on your disk. Selecting it from the Apple menu displays the window shown below. Type in the name or part of the name of the file you wish to find and click the Proceed icon (the button with the little man in it) or hit Return. *Find File* will search all your folders and display all file names containing your search characters. Preceding each file name is an icon (similar to those in the Finder when viewing by size) describing what that file is (document or application). Click on the file to "get info" on the file (size, creator and type) in one window and see its hierarchy of folders in another window.





BMUG FALL 1986

Find File can run in the background of another program in a multitasking fashion. Give it search instructions, return to your current program, and *Find File* will continue to search for your request, beeping when complete. This drastically reduces the speed of your current application, but can be useful in many situations. If you don't want to complete a search, simply click the "stop sign" icon to abort. You can also begin a search at a given location or folder to shorten search time.

Find File is free to purchasers of the new 20SC. The rest of us may have access to it in a fashion similar to the System and Finder updates, but distribution plans aren't settled yet.

Disk Info (v. 1.43)

Disk Info, by Maitreya Design, is the oldest of the three programs, having been around nearly as long as the Mac. The program was written by Dave Dunham, who wrote the Desk Accessory outliner Acta. Although *Disk Info* has been around a while, it only recently acquired the ability to find files.

The "Find" function of *Disk Info* is rather sparse, but it does work. Unlike *Find File*, *Disk Info* has no icons in its dialog box and can't work in the background, but searches for files based on the "Full Name," "First Part" or "Any Part" of the file name. *Disk Info* is faster than *Find File* by a few seconds, but cannot abort a search.



Disk Info's "Find" Window

Disk Info is not primarily a file finder, but a hard disk management tool that allows the user to delete, rename, and get info (size, creator, type) regarding a file, folder, or disk. It also can compact the Random Access Memory, sort files and folders, and view file size in characters or "k" of memory. *Disk Info* does not have the best filefinding capabilities, but compensates with other virtues, and all for \$10.00.

HFS Locator Plus

HFS Locator Plus is probably the most advanced and fastest file finder of the three, using excellent graphics and taking full advantage of the Macintosh interface.

HFS Locator Plus allows you to create folders, launch and move applications, and rename, copy, and delete files. Click on the Show Path icon, and a menu drops down showing the



heirarchical path to the file. You can also create a text file catalog and put all the findings from a search into that file. Besides searching by name, you can search by date of creation or modification, find files for the current application, find files for a different application, search for invisible files, and the program provides you with helpful hints.

One drawback with *HFS Locator* is that it cannot search for any part of a file name the way *Find File* and *Disk Info* do. *HFS Locator* will only search based on "Begins with" or "Matches" (not "Includes" a specified string). It is otherwise a very nice locator with the ability to launch a totally different application at the close of the current application (which is faster than going through the Finder).

File "MacWrite 4.5 " is in "WORD PRCS".


Show path

Cancel

Close

Dispositions

Rename

Move

Delete

Copy

Show info

Launch

Create folder

Catalog

Find next

Locator Search Status

% Searched:

Matches found: 1

Suspend search

HFS Locator Plus

Conclusion

Each of these programs is useful and inexpensive, and each is unique in its abilities. If you're need a simple locator that works in the background, get *Find File*; if you need an inexpensive Desk Accessory that has more features than a simple locator, you can't beat *Disk Info* at \$10.00. *HFS Locator* is the most comprehensive of the three and has some nice features, for which you pay more. Any will find a given file, and all three work. If you're lost in the catacombs of your hard disk, put one of these to work and find your way out.

Find File
Apple Computer, Inc.

Disk Info (v. 1.43)
Maitreya Design - ShareWare
BMUG Disk #11

HFS Locator Plus
PBI Software



Koala KAT-ing Around the Classroom

Kenneth William Hirsch

If used as part of a Macintosh classroom teaching system for in-class demonstrations, installed on a roll-around audio-visual cart, the mouse presents problems:

- (1) Lecturers frequently must move away from the cart to write on the blackboard or to check their notes. While away from the cart, it is not possible to use the mouse; the lecturer must defer manipulation of the computer until he returns to the working surface of the cart top.
- (2) With the Macintosh computer, its cables and disc drive placed on top, most carts become cluttered; mouse movement becomes circumscribed and subject to interference.

The Koala "KAT" alternative to the mouse, a stylus or finger-operated pressure tablet, seemed likely to offer a solution, as:

- (1) it would be possible to carry the KAT to a black board or other position and, when needed, move the cursor or Mac screen image around and initiate commands to illustrate the presentation.
- (2) when on the cart, the KAT would do well at defending its own territory against clutter.

After a month-long evaluation of the KAT for such use, I conclude:

- (1) The KAT remains the only convenient cheap, hand-holdable, device capable of positioning the cursor and commanding the Macintosh. As such, it would work in place of the mouse to enable use of the Mac mounted on a cart while lecturing in a classroom.

- (2) The KAT, as currently made, presents some problems:

- a. Use of the finger or stylus in its quick, direct mode results in variable, jagged lines and cursor placement.

Clicking on the smoothing function reduces the variability to make the KAT usable, but in doing so, introduces lag time which results in cursor overshoot; the user moves her finger across the plastic surface of the KAT and stops where the cursor should stop -- but the cursor goes a bit too far.

It seems likely that the smoothing function introduces a delay into the circuit which acts to draw on-screen the changes in stylus/finger position as they occur on the KAT surface. The delay would be coupled with a moving average of a variable number of stylus/finger position data points, to "smooth" the cursor on-screen. The problem is that the more the smoothing, the larger the number of data points averaged and the greater the "inertia" of the stylus movement. Koala needs to create a more sophisticated design.

- b. Stylus use is O.K. only when movement is crisp. When movement is minute and slow, as is required in detailed drawing, the KAT "skips" a great deal.

It is probable that more experimentation with settings of sensitivity and smoothing could improve the situation, but each user would have to spend time with the KAT to find her own best



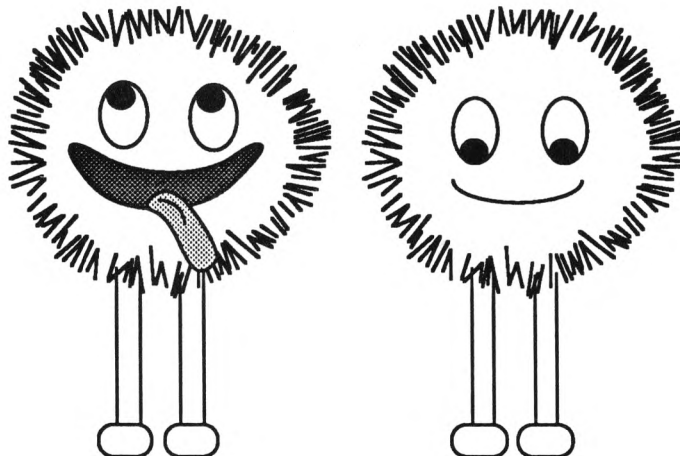
settings, thus invalidating KAT usage with stylus in participatory classroom settings where there is no extended opportunity for each user to make trials and adjustments.

c. the switching function which is the equivalent of mouse "drag" is inconvenient.

Nevertheless, despite the problems mentioned, KAT is currently the best way of dealing with the lack of mousing surface associated with using a portable, cart-mounted Mac system to accompany classroom lecture/student interaction with the Mac computer system.

The "KAT" system is available at a list price of \$199, but can be obtained from typical discount supply houses for \$135 plus shipping. Information may be obtained from:

Koala Technologies
2065 Junction Avenue
San Jose
California 95131
phone: 1(800) 562-2327



Puffs/Dan Wood

Micrograde and the Joys of Teaching Undergraduate Classes

Jasper Rine

Of the many joys of teaching undergraduate classes at a large university, the bookkeeping involved in grading is not one of them. Even in the spoiled and pampered position of having graduate teaching assistants who readily and cheerfully grade exams, and enter and compile the data, I still find that the basic job of maintaining grade records excessively time consuming, utterly boring, and since it is boring, prone to errors. These errors tend to be the clerical type that are difficult to track down, and capable of causing high anxiety among the students. It was inevitable, at a time when there are Macintosh programs for tasks as mundane as balancing a checkbook, that someone would write a program to handle grade bookkeeping and calculations. *Micrograde*, by Chariot Software of San Diego, is a recent arrival in this field. The acid tests that need to be applied to all such "work-saving" programs are (1) Does the program take more work than the way you were doing the job before?; (2) How badly can the data be screwed up by bleary-eyed teaching assistants or professors?; and (3) Are the results worth the effort to anyone other than a Macaddict? In my opinion, *Micrograde* comes close to filling the bill. Before I discuss its virtues and few limitations, I will describe the teaching situation that I deal with in order for you to understand the prejudices in my evaluation.

I team teach a large undergraduate class that typically has 150 to 175 students enrolled. The grade each semester is based upon the cumulative score on three hourly exams and a final exam. Presently, we use programs run on a VAX-Unix operating system to record and process grade information. Thus, each semester a teaching assistant volunteer has to manually enter student names, student numbers, and test scores into this program according to the rights and rituals of Unix. We have a resident guru who patiently explains the Unix voodoo to each years' teaching assistants. Unfortunately, our guru will soon earn his degree and leave us to either adopt new software or resort to our pocket calculators. *Micrograde* passes the first test easily, since entering the data in *Micrograde* is no worse than Unix and it is much easier to edit errors in *Micrograde*. One slight complaint that I have is that it is possible to enter the same name twice without raising the program's eyebrows. This type of error could lead to significant clerical problems later if it isn't corrected.

The data and calculations appear in four tiled windows that present different types of information about the class. One window is a **Class Roster** that can list the students by any criterion that one wants. The second window is the **Student Summary** window. When a student is selected from the roster window, the data concerning that student's entire performance in the class is presented in this window. The third window is the **Class Scores** window. This window contains a class roster where the performance of the entire class can be displayed for any assignment by simply scrolling through the possible assignments with the scroll bars at the top of the window. The fourth window is called **Categories and Assignments** and it is in this window that the grading structure of the course is defined by the instructor in response to simple menu formats. The visual effect of the windows is very clear and intuitively organized and represents all aspects of classroom bookkeeping. The windows exploit the Mac's visual approach to program operation and should be self-evident to anyone who has any experience with other Mac software.



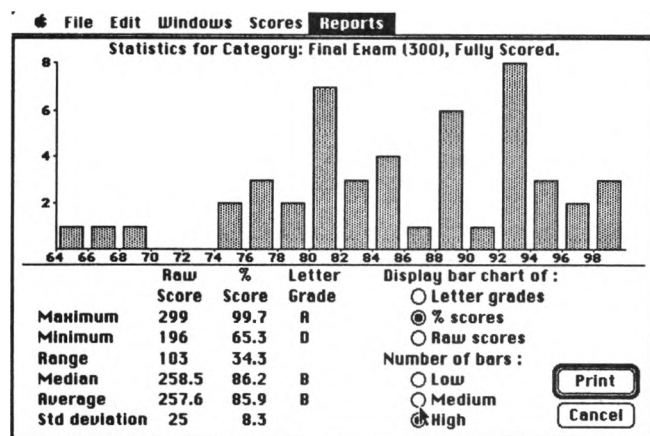
File Edit Windows Scores Reports													
☐ Roster for test db													
Class has 44 students. Total points: 68													
☒ Name	Student Summary Jones, Reese I.D. 72729. Grade Average: 558, 82												
Hambrecht, Betsy Heckendorn, John Huxham, Fred James, Jay Jones, Reese Kane, Beverley King, Tony Kowalski, Nicole Kronick, Scott	<table border="1"> <thead> <tr> <th>Category</th> <th>Assignment</th> </tr> </thead> <tbody> <tr> <td>Midterm1 (100)</td> <td></td> </tr> <tr> <td>Midterm2 (100)</td> <td></td> </tr> <tr> <td>Midterm3 (100)</td> <td></td> </tr> <tr> <td>Notebook (80)</td> <td></td> </tr> <tr> <td>Final Exam (300)</td> <td></td> </tr> </tbody> </table>	Category	Assignment	Midterm1 (100)		Midterm2 (100)		Midterm3 (100)		Notebook (80)		Final Exam (300)	
Category	Assignment												
Midterm1 (100)													
Midterm2 (100)													
Midterm3 (100)													
Notebook (80)													
Final Exam (300)													
Categories & Assignments													
Contains 5 categories, 5 assignments													
Midterm1 (100) Midterm2 (100) Midterm3 (100) Notebook (80) Final Exam (300)	Scores for test db Category: Final Exam (300) Fully Scored.												
☐ Name													
Heckendorn, John Kronick, Scott Kowalski, Nicole Hambrecht, Betsy Takatsuka, Jim Rine, Jasper													

The bottom line in the grade crunching aspects of the software is that the program is flexible enough to do whatever you want to do in the way of calculating grades. Moreover, it is easy to get what you want. The program can do weighted averages, plain averages, delete the lowest n scores, and accommodate such variables as incompletes, dropped students, and excused absences. There are more options available for handling grades than one would probably use in any one class, but at the same time the program is not weighed down with marginally useful frills that would make it more difficult to learn. With regard to the second test, I spent several hours one evening trying to foul up the entry of data. The program is remarkably resistant to stupidity. For example, I could not enter more points on an exam than the exam was worth without having the program warn me. Nor could I impose an illogical cutoff for grade standards (i.e., it is impossible to establish a B as greater than or equal to 80% and then set a C as greater than or equal to 85%). The only errors that I could get away with were to put the same name in twice, and once I managed to erase the entire class roster. The program has a built in security feature that places an asterisk next to any score that has been changed after it was first entered.

My one serious complaint has to do with how the curve for a course is established. In this program, one chooses the score cut-offs for each letter grade (pluses and minuses are accommodated) and then the program calculates the final grades based upon the cumulative score or upon individual grades. If the curve does not look good, one can simply change the cut-offs and the program recalculates the grades and the curve. For many applications these approaches will be sufficient. However, in our large class, we always plot a curve of score versus rank with each student represented by one dot on that curve. In establishing the cut-off between a C- and a D+, for example, *Micrograde* is blind to whether the division separates students whose total scores differ by only a single point and heartlessly makes the cut. In contrast, Berkeley professors exhibit their infinite wisdom and compassion in such cases and look for

BMUG FALL 1986

the next logical breaks in the curve. If the class has been alert and laughed at all of the professor's jokes, he or she will probably choose the break on the downhill side to divide letter grades. Hopefully, the next release of *Micrograde* will incorporate some humanity and make this option available.

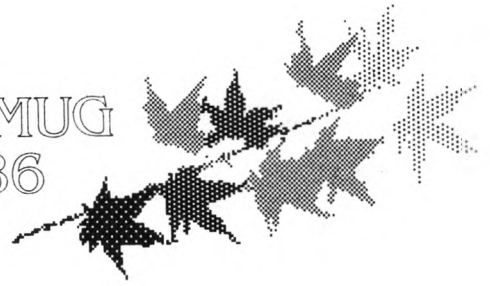


The last two features are two of the strongest points of *Micrograde*. The instruction manual is well-illustrated and lucid, although a bit redundant. If DOS software documentation had been written this well, I may never have tried a Mac. The most significant strength of this program, in my opinion, is the variety of ways in which printed reports of the data can be prepared. The program will print histograms of individual assignments or test scores and list the class either by name or student number with the corresponding grades. Alternatively, for courses such as ours in which a grade isn't calculated until after the final exam, the reports can list the students and their relative rank in the class on an individual exam, or in the entire class. The ability to obtain this type of data easily answers the final question of whether the effort was worth it. Given that our universities have complex rules of when a student can drop a course and not have the drop appear on a transcript, or options that sometimes let them repeat a class to change certain letter grades but not others, it is only fair that they be given as complete information as possible regarding their relative progress. *Micrograde* is a tool that should help in this goal, and I intend to try the program under real battle conditions this year.

Jasper Rine is on the faculty of the Biochemistry Department at the University of California, Berkeley. Besides his Macintosh, he enjoys yeast genetics and fly fishing.



balloons/Wood



All the Home Accounting Programs

Scott Beamer

What will a home accounting program do for you? Maybe it will help you clean up your act, and when it comes to money, almost all of us need to clean up our acts. It will help you balance your checkbook and keep it up to date. It will help you balance your accounts from your monthly statements. It will help you make, and stay, on a budget. It will remind you which bills you meant to pay when. It will even write checks for you (and address the envelopes if you use windowed envelopes). Best of all, it will be a big help at tax time. It will probably save you more than the cost of the program the first year. By the way, "Home Accounting" is a misnomer. These are actually "home bookkeeping" programs. If you want an accounting program, there are literally dozens to choose from, but they assume accounting skills to use them. If you can use a Mac, you can use any of these programs.

MacMoney (Survivor Software) \$79.95. Version 2.02. It is the current home accounting program of choice. It has the most elaborate features of any home accounting program on the market. Data entry is fast and intuitive, with separate forms for checks, credit cards, bank charges, and even adjustments. There is a memo line on all these forms, so you can leave a good paper trail to remind you later what you are doing now and why.

Report generation is extremely elaborate, allowing the user a maximum of customizing capabilities. There are so many choices, it's intimidating to use, but you will not need to generate reports until you have gotten used to the program for a while. By then, it will be easier. There are added features, such as mortgage analysis, IRA planning, and retirement planning. I find I hardly use these features, though I don't feel I paid much extra for them. The mortgage analysis is impressive, though and generates a complete thirty year amortization table.

MacMoney just got handier. **MacIntax**, the personal income tax program, has just announced that it will be able to import text files this year from many programs, including **MacMoney**, and allow you to mouse click the numbers into your tax forms. Next year, **MacIntax** expects to have this process automated so that **MacMoney's** annual report will automatically fill out your forms. Sound too good to be true? Yeah, but I got to try it. This program is fine for students, yuppies, hackers, families, in fact, most anyone. If you have a small business, charity, or user group, this may be all the accounting software you need.

The screenshot shows the MacMoney interface. At the top is a menu bar: Apple icon, File, Edit, Transactions, Lists, Plan, Reports, Graphs. Below the menu is a 'Names' list on the left with entries like 'McCall's', 'Montclair Children's Shop', 'Monthly Interest', 'Neiman-Marcus', 'Nordstrom', 'Northwestern Mutual Life', 'Oakland Seavenger', and 'P.O.A.E.'. The main area is a 'Scott Check' form with fields for Date (06/28/86), Balance (3,814.25), Check # (806), Pay To (Neiman-Marcus), Memo (Can't Afford!!!), ID Code, and Category (Merchandise). There are 'Save', 'Quit', and 'Split' buttons. At the bottom, there is a 'Categories' list with 'Type' and 'ID Codes' sections.

MacMoney

The screenshot shows the Dollars and Cents interface. At the top is a menu bar: Apple icon, File, Edit, Select, Maintenance, Report, Graph, Controls. Below the menu is a 'BEGIN NEW ACCOUNT FILE' dialog box. It has fields for Title (Joint Accounts), Year (1986), and First Month (1). There are 'OK' and 'Cancel' buttons. Below the fields are two sections: 'Initial Set of Accounts' with radio buttons for 'Tax Preparation Accounts', 'Household Accounts' (selected), 'Business Accounts', and 'No Initial Accounts'; and 'Accounts Selected' with a list box containing 'Mortgage', 'Check Interest', 'Dividends', 'Gross Income', 'Interest Income', 'Auto Expenses', and 'Check Charges'.

Dollars and Cents



BMUG FALL 1986

Dollars and Cents (Monogram Industries) \$150/\$93. It is now out in version 1.4. This version is HFS compatible, has a mini finder and no copy protection. Upgrades are available at additional cost to registered owners. It opens and closes faster, but has no other apparent differences from earlier versions. This program has been among the most popular home accounting programs. It has sold over 100,000 copies in versions for various machines. I used it for a year, but what is needed is version 2. This is rumored to be available sometime next year.

In the meantime, Monogram has been busy with other projects. They have introduced a companion program for *Dollars and Cents* called *Forecast* that offers increased graphic and forecasting abilities. They currently have in place some hookups for home banking, but none in California yet. They hope to soon offer a tie in with a tax package so that your end of the year report will be automatically transferred to your tax sheets.

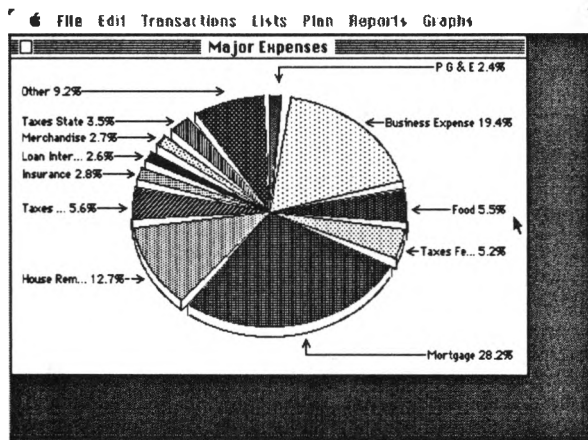
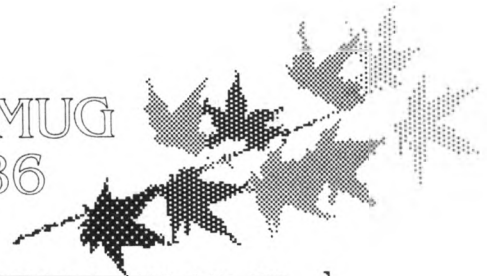
In terms of performance, I'd wait and see what version 2 is like. My biggest gripes about version 1.4 are the same as with version 1.3, that there is no memo line on checks and its report generating abilities are limited. I consider the memo line mandatory if one is going to leave an adequate trail of notes to remind one why one did things when looking back from a year or years later. Hmmm? Why did I write a check to Miss Toots LaRue for \$200? Look at the memo line and see "Party Favors". Oh, that's it!

The reports are limited and offer inadequate editing capabilities. The part that irks me the most is that the budget features can't be turned off. I don't happen to use the budget features, so I find this very irksome, causing lots of blanks and spurious data in the reports and charts. For the impatient, you can buy version 1.4 now and upgrade later. To give this program a personality, I would say this one most feels as if it were written by an accountant. It's certainly reliable, but unimaginative... the manual, too.

The Home Accountant and Financial Planner for the Macintosh (hereafter to be referred to as *HAMac*) (Haba Systems) \$115/\$64. This is a product dead in the water. The programming team that wrote it no longer has a working relationship with Haba. There are no further upgrades for the product planned and no bug fixes. Though the program has some bugs, it actually works quite well. I prefer it to *Dollars and Cents*. The only reason I don't use it is that there is a better product available. The programming team that wrote *HAMac* have formed their own company and are offering what is essentially Version 2 (2.02 at last count) of the program. It is now called *MacMoney*.

The improvements of *MacMoney* over *HAMac* include: it will import *HAMac* data (but *HAMac* will not import *MacMoney* data), it invisibly looks for some data base bugs and will attempt to fix any found every time you open a file, and it will print out reports in ASCII text. This allows you to move reports to a word processing article, or to *Excel* for further processing, or to *MacIntax*.

ComputerWare includes a book on home financial planning (not Mac specific) for the \$64 price.



Home Accountant and Personal Financial Planner

Date	Num	TDate	Account	Type	Amount	Balance	Destination	Description
01/08			AMEX	PAYMENT	325.00	325.00		PAYMENT
01/20			AMEX	PAYMENT	35.00	55.00		PAYMENT
01/02			AUTOLOAN	PAYMENT	210.00	210.00		PAYMENT
01/20			AUTOLOAN	PAYMENT	210.00	210.00		PAYMENT
01/15			CHECKING	EXPENSE	6.00	6.00	BANK CHG	
01/20			DINER	PAYMENT	30.00	30.00		PAYMENT
01/11			NC	PAYMENT	100.00	100.00		PAYMENT
01/20			NC	PAYMENT	25.00	25.00		PAYMENT
01/02			MORTGAGE	PAYMENT	590.00	590.00		PAYMENT
01/30			SAVINGS	INTERMD	6.00	6.00		INTERMD
01/03	104	01/10	CHECKING	CHECK	33.90	33.90	FOOD	Albert
01/01			MORTGAGE	OPENBAL	80,000.00	80,000.00		OPENBAL
01/02			NC	EXPENSE	131.00	131.00	AUTO	Earl I
01/01			AUTOLOAN	OPENBAL	4,000.00	4,000.00		OPENBAL
01/15			DINER	EXPENSE	30.00	30.00	FOOD	Maxwell
01/01		01/01	CHECKING	DEPOSIT	332.91	332.91		DEPOSIT
01/01			NC	OPENBAL	250.00	250.00		OPENBAL

OverVUE Personal Finance Template

OverVUE Personal Finance Template (ProVUE Development Corp.) \$39.95/\$34.95. It is very inexpensive, and if you already have and like *OverVUE*, this may be an adequate solution for you. The template has almost all the features, but is not as agile, as a pure home accounting program. This is somewhat offset by the fact that *OverVUE* will allow the advanced user to customize these templates to their heart's content. For that person, this program provides great power, maximum customizing abilities, and ease of use. A word of warning... *OverVUE* recommends that you do not try to customize the structure of the template as it will interfere with the Macros and hence, the operation of the program. If you attempt it, do it only on an expendable copy.

This program also lacks the Memo line on checks. The reports are clumsy to implement and slow. There are charts available. Data can be exported to other programs as a text file, for instance, *MacIntax*.

Electric Checkbook (State of the Art) \$79.95/\$39 This is the junior member of the group. It does what it does well, but it lacks the features of the other programs. This might be the program of choice for someone who is more concerned with how much of which bills to pay this month than say, retirement income planning.

The basics of the program are well implemented. Error correction is especially good, but the manual is only 44 pages...with a fourteen page addendum...with a one page addendum. The fourteen page addendum covers such items left out of the manual as credit cards and mortgages. The one page addendum tells: of a new feature allowing checks to print the payees address (a feature the other programs have except *OverVUE*), the upgrading of the finder to 4.1, that **version 2.0 will not accept data from the previous versions**, (all version 1.0 data must be rekeyed (Argh!!!)), and finally, all tech support is by mail only. To me, it appears that State of the Art (that's their name, honest... and based on these features, a sorry state it is!) has spent more time worrying about **copy protection** and keeping you from xeroxing the manual (it's printed in grey and blue) than paying attention to the value of their product, especially compared with the competition.

BMUG FALL 1986

Clr	Check	Date	Amount
✓	Wl th	02/19/85	1,000.00
✓	Dep	02/28/85	1,100.00
✓	Adj	03/05/85	15.00
✓	Dep	03/15/85	900.00
✓	BrkCg	03/16/85	10.00
✓	Adj	03/22/85	132.32
✓	BrkCg	03/22/85	6.00
✓	Wl th	03/27/85	75.00
✓	Int	03/31/85	23.94
✓	Dep	04/01/85	1,250.00
✓	Wl th	04/22/85	32.00
✓	Wl th	05/01/85	50.00
✓	00100	03/25/85	79.45
✓	00101	03/01/85	225.00
✓	00102	03/15/85	0010
✓	00103	03/28/85	20.00
✓	00104	03/31/85	25.00
✓	00105	04/12/85	12.00
✓	00106	04/12/85	23.83
✓	00107	04/15/85	2,118.00
✓	00108	04/14/85	464.32

Statement Balance: 5,025.10

Check Number:

Outstanding...

Checks (-): 2,675.99

Deposits (+): 1,250.00

Adjustments (-): 82.00

Net Balance: 3,517.11

Current Balance: 3,684.11

Out of Balance By: -167.00

Accept Report Purge

Enter your name and address as it should appear on your reports:

Name: J. K. Lasser.

Address: 2160 Stoned Road

Berkeley, CA 94705

Report Page Length: 60 lines.

Number of Accounts: 6 Number of Budgets: 7

Number of Payments: 1 Number of Deposits: 1

Number of Charges: 1 Other Transactions: 1

Available Memory: 212,274 Available Disk Space: 7,971,840

Electric Checkbook

Your Money Manager

J. K. Lasser's YOUR MONEY MANAGER (Simon & Schuster) \$99.95. This program was released this week. It was received literally too late to review. I have had about two hours to fiddle with it, and based on that time, and reading the manual, here are my impressions.

The program has two big selling features. One is that J. K. Lasser has produced the most popular personal income tax guide for almost fifty years. There is instant recognition and a great deal of respect associated with this name. The other feature is that the program is designed to work directly with a companion program, **J. K. Lasser's YOUR INCOME TAX**.

My first impressions indicate that both these features have worked against the program to some degree. In the first case, while Lasser's tax guide is printed on cheap paper in a large densely worded workbook with exhaustive indexing, **YOUR MONEY MANAGER** comes in a hardback manual with a garish oversized cover. Inside is a laughably small manual, luxuriously printed, that never gets past the first time tutorial stage of dealing with the program. There is no index! Fifteen percent of the manual are four pages of full sized screen dumps showing each of the pull down menus in the main menu bar. Any but the most timid of Mac users will be impatient if not insulted with the spoon feeding style of this manual.

The second feature, the connection to **YOUR INCOME TAX**, causes another set of problems. The idea is to allow you to shift all you bookkeeping data from **YOUR MONEY MANAGER** to **YOUR INCOME TAX** with the click of a button, thus filling out your income tax forms almost instantaneously. I have not entered enough data to try this out, but already I have found the principal drawback to such a scheme. Your personal files must be arranged according to the precise list of categories needed to effect the transfer. Other than listing the categories along with an I. D. code, the program does not help with this.

Worse yet, the program requires you to enter a four digit code for every transaction. There are dozens of these categories to memorize. You can have a screen of the codes visible, but there is no implementation of a double click data entry system that some of the other programs have.

Don't get me wrong. The package looks good and I'm certain it works. It is written in MacForth. It has nice windows and controls, and is impressive in its speed. It is fairly full featured, and has lots of command keys. My favorite is Command-E that closes all files and does a complete shut down. The rest of the programs should take note of this one!



It will print checks (free samples enclosed), make automatic payments, print charts, and do several reports, including a personal financial statement. There is tech support by phone, and it is not copy protected.

My first impression tells me that this has the basis of a nice accounting-bookkeeping package, but version 2 is needed right now.

For a fuller evaluation of these products, see October Macazine.

Products:

MacMoney
Survivor Software
11222 La Cienega Blvd.
Inglewood, CA 90304

Dollars and Cents
Monogram Industries
8295 S. La Cienega Blvd.
Inglewood, CA 90301

**Home Accountant and
Personal Financial Planner**
Haba/Arrays
11223 South Hindry Ave.
Los Angeles, Ca 90045

Personal Finance Template
OverVUE
ProVUE
222 22nd St.
Huntington Beach, CA 92648

Electric Checkbook
State of the Art
3191-C Airport Loop
Costa Mesa, CA 92626

YOUR MONEY MANAGER
Simon & Schuster
1230 Avenue of the Americas
New York, New York 10020

Objective Data

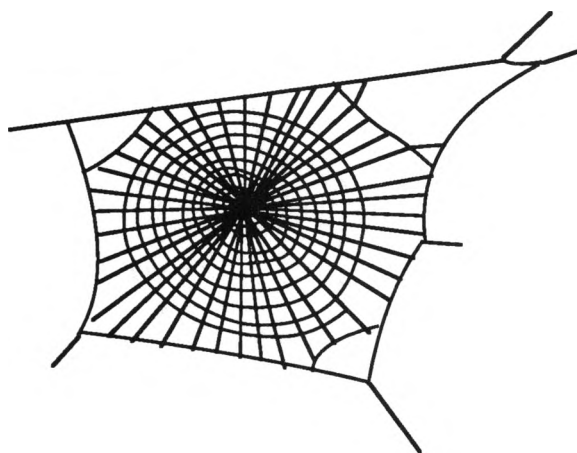
	MacMoney	HAMac	Dollars & Cents	OverVUE	Electric Checkbook
List Price	\$74.95	\$99.95	\$150.00	\$39.95	\$79.95
Discounted	?	\$62.00	\$81.00	\$34.95	\$42.00
Minimum System	512k	128k + ext	128k + ext	128k + ext	128k
Num of accounts	200?	200?	12	limited	10
Num of Categories	200	200	120	by RAM	100
Num of transactions	3000 to 6000	3000	2000	3500 ca	500
Size of Program	255k	208k	228k	129k*	60k
Copy Protected	no	no	yes	no	yes
Help Menus	yes	yes	yes	no	yes

* will work only with OverVUE 2.0 which is another 145k

Subjective Data

(highest rating = 1)

	MacMoney	HAMac	Dollars & Cents	OverVUE	Electric Checkbook
Easy to learn and use	3	3	4	2	1
Follows Mac Interface	2	2	3	1	4
Flexible	1	2	4	3	5
Full Features	1	2	3	4	5
Report Quality	1	2	4	3	5
Charts Quality	1	1	3	2	none
Speed	5	4	1	2	3
Technical Support	1	2	4	3	5
Manual	1	2	3	5	4



Wood/SpiderWeb



Mini Reviews:

Why Don't They... Hey, Wait -They just did!

Scott Beamer

MacInUse. That's exactly how it happened. I was talking to Softview on the phone, and they mentioned their soon to be released new program. Exactly the program I had just described in a "Why don't they..." for this newsletter. Picking myself up off the floor, I asked for a copy; and now I am back at my Mac, ready to tell you about it.

Remember Softview? Name ring a bell, but you can't quite place it? You'll know the name again in a few months. The Softview people produce MacInTax. Their new program isn't for taxes. Though you could use it for tax records. It's called MacInUse. It will list for \$49. They're hoping to release it at the Dallas Expo. It's a desk accessory. It installs itself in your system through an INIT. Don't worry about that. You just click the icon; it does the rest. MacInUse creates a time log for the period your Mac is on, recording how long you use each application.

I can't say how many people have the same compulsion about time that I do. To me, this program seems like a sure winner. With it, you can produce records to justify how you depreciate your computer expenses, or to do client billing. You can see what your kid was up to while you were out last night. This is a general purpose tool; figure out the use that suits you. It will certainly be tax deductible.

MacInUse shows the intelligence and careful crafting I have come to associate with MacInTax. It can be turned on and off without removing the application. You can save a file, and start a new one, resetting all the clocks. In visible mode a dialog box allows annotation of the file, with buttons for Business or Personal, and a space for text. Alternatively, the police functions can be enhanced by making the program invisible, so no one else will know it's there. It will print out its files. It comes with templates to open the files in MacWrite or M. S. Word. It has a macro for Excel that will graph your time use from the MacInUse file.

A few day's use has revealed some shortcomings. It is slick, but I would like MacInUse to log the time under individual file and document names rather than just the application name. I also think the dialog box should look like the file does and should include a display of the time just logged. On my hard disk, it even logs my boot up disk. This is awkward as it requires clicking away the dialog box that would normally be used to annotate the MacInUse file. Still not perfect, but what a great program idea!

MacInTax (Softview)

It's getting towards that time again. Tax time. April 15th. MacInTax remains the only tax program worth considering, unless your needs are very modest, or you like to tinker with Excel templates. Susan Morgan, co-owner of Softview, promises the final release for 1986 MacInTax for the San Francisco Mac Expo in early January! They guarantee not to be late this year. A prerelease update will be available, but only to registered owners, in November.)

Some changes are in store. The main disk still lists for \$99.95 but will now include all the federal and business forms. The Business Supplement is eliminated. Updates will be \$45, on special this year for \$30; California forms are an additional \$45, (\$15 update).

There will be converter routines on all disks. The federal converter offers two options. One will convert the permanent information, such as name and Social Security number, from the 1985



forms to the 1986. It also transfers and updates Schedule G (Income Averaging). The other option will convert all information on the 1985 forms so that the 1986 MacInTax can read the files.

The big news for this year is that MacInTax will accept text files from Excel, Jazz, many full accounting programs, whatever. It works especially well with MacMoney; it does not work, this year, with Dollars and Cents. You will see your transferred files on screen in a window. Select some data, then point on the tax form to where you want it and click to complete the transfer. In 1987 MacInTax will do this transfer automatically with selected programs including MacMoney, Dollars and Cents, and several other large accounting programs. A professional accounting program from MacInTax is also in the works.

MacZap 4.3 & Lifeboat.

Micro Analyst has just done an extensive workover of their MacZap package. Their ideas seem good, but the results are disappointing. Everyone I know reaches for Copy II Mac when they need a copy, even when MacZap Copy is available. I don't expect to see this change, as MacZap hasn't changed much in essence. There is something very unMac-like about the screen and controls.

The utilities now handle HFS and 800K disks. Fast Disk Copy is a new addition to the disk that facilitates fast disk copying, especially if more than one copy is being made. It requires some disk swapping.

Micro Analyst has a version 2 of their book on software protection (\$33). This book is not specific to the Mac. There is a new book on the details of the difference between MFS and HFS (\$18). I haven't seen either of these books, but if the MacZap manual is any example of their writing style, look out! The manual is one of the most opaque I have ever tried to read.

Lifeboat is a BBS run by Micro Analysts for registered users of MacZap ((512) 926-9582). It is apparently intended to allow registered users rapid access to new patch files without having to wait around for the next update of MacZap. The BBS includes section for: File, Floppy Disk, and Hard Disk recovery, Patch File Downloads, Developer's Information and Support, Public Domain Programs, and Messages.

A good idea: but results remain to be seen. The BBS started up in August, but no one is there. I have called in several times over six weeks checking for new files or messages. In all that time, there was only one new file or message left. My messages to sysop were never answered. The patch file list remains exactly the same as the one with the latest MacZap upgrade disk. (Save for some minor patches to fix bugs in the various MacZap utilities.)

Why Don't They?

Does anyone know of a database mailing list of the United States Congress? If there isn't one, why doesn't BMUG make it a public service project to make one? It's only about six hundred names. We could split up alphabetically for typing, and later join the lists, if we do it in OverVUE. The information ought to be in the public library. It might become BMUG's most popular disk, a good attention getter at shows. It would be nice on a BBS, as well. I can think of some very good uses for this. Can you? Ought to work great with M.S. Word mail merge. (See Piracy and Rentals).



through and trace your program, set stops, observe variables on the fly, and support text and drawing windows automatically. These types of tools are usually only available to programmers working with a slow interpreter. They are rare in a compiler. *LSP* also supports the concept of a Project, so you get SmartLinking and compact code automatically without having to consciously manage various files.

If you are familiar with the *MacPascal* interpreter (also by THINK), you will feel at home with *LSP*. Most *MacPascal* programs will compile immediately in *LSP* with the addition of only two commands: ShowText and ShowDrawing. If you have been using *MacPascal* up to this point, you will probably want to switch to *LSP* to gain compiled speed while your programs are running.

Nothing is perfect, so of course *LSP* has some flaws. If you are writing programs for yourself or are not overly concerned with the size of your application, *LSP* will suit you fine. It will shorten enormously the time it takes you to write a bug-free program. However, if you are writing a program to sell commercially, or if disk space is precious, you will probably want to develop your program in *LSP* and then recompile a working version with either *TML Pascal* or Apple's *Macintosh Programmers' Workshop Pascal*. Compiling the finished code with another compiler may save you a few K in the finished product and may make your program run just a little faster. In general, *LSP's* rules are a subset of the rules used by other compilers, so moving code out of *LSP* should be painless. Moving code in to *LSP* might take a little more work.

If you are developing a very long application, you may run up against *LSP's* limitations on code length. *LSP* encourages you to write your code in *units*, and no unit can be longer than about 2000 lines. Since an application can have a very large number of units, this requirement should not keep you from writing a program of the complexity of, say, *Excel* or *Omnis3*, but you may have to juggle some code around occasionally. If you can be disciplined enough to write all your code in modular units, you'll find that recompiling will be much faster than if your whole program is in just a few units. *LSP* is smart enough to only recompile what has been changed.

The Animator

If you need a little help to get started programming, take a look at a small program I put together in *LSP*. The source code and the compiled program for this article are available on the BMUG Fall 1986 membership disk, so you can take a look at the necessary resources using *ResEdit* and you won't have to type in the code yourself. (*Important:* Without the appropriate resource file, this program will crash and burn when you compile and try to run it!) I call it *The Animator*, and its sole reason for existing is simply to take (almost) any number of previously-stored 'PICT' resources and animate them in a window that can be dragged but not closed or resized. It doesn't do much, but it illustrates the concepts of a main event loop and drawing in windows. It also illustrates how to store strings, menus, and pictures as resources and load them from your program. *Animator* supports DAs (as any good application should—you'll see that it's not hard), and gives menu options for quitting and reversing the animation order. You can use the *Animator* source code as a shell, cutting a little of the code, and adding a lot of code to make a program that does whatever you like.

The structure of *Animator* is shown in Figure 1. Normally, with a program this small, you would be able to include all the necessary code in just one file. I have separated this program into six files, however, to illustrate how units would be used for more complex programs.

Unit by Unit

As soon as you begin to examine any *LSP* source code, you'll note that it follows the same structure that most modern Pascal compilers do: that of **units** made up of **interface** and **implementation** sections. Whenever you want to include a **unit**, you say a **program** or a **unit** **uses** another **unit**. The idea of **units** originated with UCSD Pascal, and has gained widespread acceptance. So with *LSP*, all projects that generate stand-alone applications have only one **program** but an almost unlimited number of **units**. Some projects like desk accessories and 'FKEY's may be made up entirely of units only.

The first unit in *Animator* is not a unit at all—it is the main program. It simply initializes a few things and falls into the main event loop. It has no "clean up" routine because we don't create any files that we need to save for next time or modify any system globals. The next unit, *UGlobals*, has **const**, **type**, and **var** declarations for identifiers used throughout the program. As with any good Pascal program, *Animator* uses numbers sparingly throughout the code so that parameters can be changed easily by changing their global definitions.

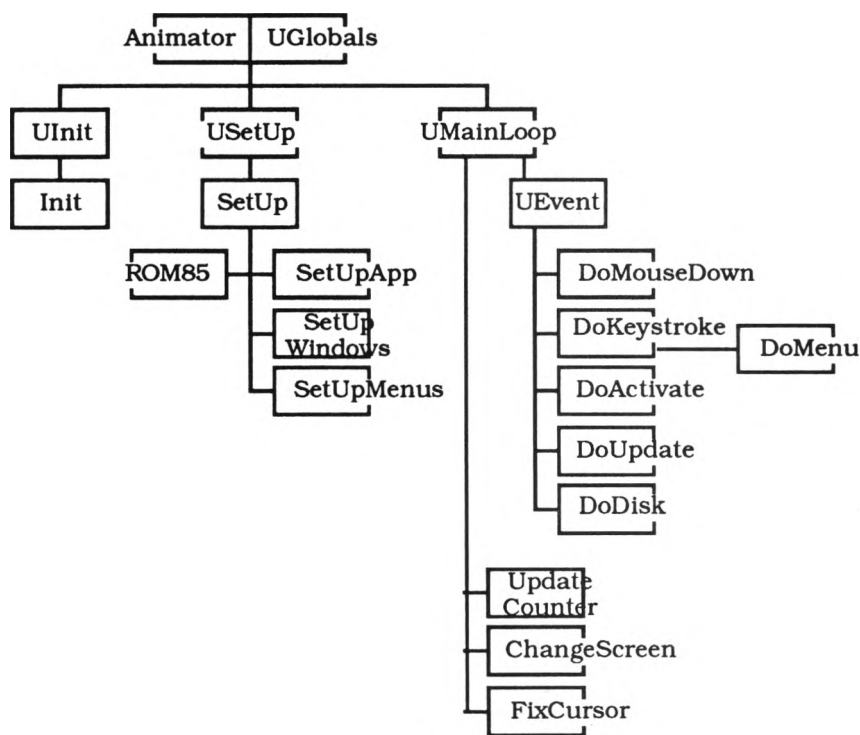


Figure 1. Schematic of Animator

The next unit, *UInit*, notes as a comment that *LSP* automatically performs the Macintosh equivalent of a "witch chant" at the beginning of every program (unless you explicitly tell it not to). The "witch chant" initializes the Toolbox. If you compile this program with *TML* or another compiler that doesn't do Toolbox initialization automatically, you should make the bracketed comments real Pascal procedure calls. *UInit* then calls **MoreMasters** four times. Since the Macintosh's memory management scheme uses master pointers to point to almost everything, and since the "things" can be relocated at any time, their master pointers are the only things that *don't* move. By calling **MoreMasters** at the very beginning of our program, rather than waiting until the program needs more master pointers later, we put all the master pointers that we will need (and some extras for desk accessories to use) at the very bottom of our



heap. We thus avoid placing chunks of immovable master pointers in the middle of our application heap. Each call to **MoreMasters** allocates a block of 64 master pointers. Lastly, Uinit calls **FlushEvent** to rezero the event queue so our application won't have to deal with any events that weren't processed by previous applications.

USetUp gets the menu and picture resources we need for the rest of the program, puts the menu bar (including all the system desk accessories) and application window up on the screen, defines how far the application window can be dragged, and initializes some global variables. It counts the number of 'PICT' resources contained in only the application file (with **Count1Resources**) and loads each 'PICT' in turn into a doubly-linked ring of pointers. After it loads the first 'PICT', it checks to see that the 'PICT' was really available. If it wasn't, **GetPicture** returns a **nil** handle, so we notify the user that no 'PICT's were available, then quit. USetUp also calculates where to place the window to center it on the screen and make it just a little bigger than the size of the stored 'PICT's. I believe the window placement code will center Animator's window on *any* screen, even a Radius or E-Machine screen, but I haven't tried it yet.

If you compile *Animator* to work on a MacPlus or 512K Enhanced Mac only, leave USetUp just the way you found it and include the ROM85 library and ROM85 files. If you compile it to work on *any* Mac, then you'll have to make a change to USetUpApp. I originally wrote *Animator* to contain only definitions found in Chernicoff and the old ROMs, but I didn't want the code to specifically compile to show only a fixed number of 'PICT's. The **CountResources** call seemed to do the trick at finding out how many 'PICT's were available. However, it included 'PICT's in the system file as well as those in the application file. Only the new ROM call **Count1Resources** fixed the problem. So, for an old-ROM Mac, leave out the ROM85Lib and ROM85 files and define MaxPict as a constant (equal to the number of pictures you want to display) rather than a variable. Eliminate the **Count1Resources** call.

The Main Event

After we've initialized everything, we're ready to enter the main loop, contained, curiously enough, in UMainLoop. This is the loop that sets every Macintosh application apart from the rest and waits for the user to tell the computer what to do. First, we call **SystemTask** to give desk accessories a chance to do things they need to do. Every application should call **SystemTask** once each tick (1/60th of a second). Although calling it once each time through the main event loop is usually sufficient, if your application is doing something that will take a long time inside of the main event loop, you should add **SystemTask** calls there as well. Next, we call **ChangeScreen** to invalidate the application window. Rather than doing the drawing within the main event loop, invalidating the window leads to an update event that will redraw the window later. Drawing the window immediately might be more expedient in a simple program like *Animator*, but generally drawing is better left completely to update events. Once more than one window is doing more than one thing, code can get tangled quickly and update events simplify things enormously. **ChangeScreen** also updates the handle to the picture to be drawn by traversing the linked list of available 'PICT's.

Now, we're ready to check with the operating system to see if we have to do anything. A call to **GetNextEvent** returns a boolean value: TRUE if an event is waiting, and FALSE if no events are waiting. Since we invalidate the application window every time we go through the main event loop, we'd better take care of that update event first. Otherwise, we will always have events waiting in the queue that we'll never process. So, we call **GetNextEvent** with a mask that only fetches update events. We pass the message from **GetNextEvent** to our update procedure. Then we call **GetNextEvent** again, this time with a mask that fetches all events. If the result is true, we use a **case** statement to process the event. Lastly, we call **FlxCursor** to find out whether the



BMUG FALL 1986

mouse has been still for at least TickWait ticks. If the mouse has been still, we call **ObscureCursor** to hide it until the mouse moves again.

UEvent

```
begin { DoUpdate }  
  TargetWindow := WindowPtr(MemTreat(msg->destWindow));  
  if TargetWindow = nil then  
    begin  
      { ; Update events should be handled by the target window  
      BeginUpdate(TargetWindow);  
      DrawPicture(Picture, TargetWindow);  
      EndUpdate(TargetWindow);  
    end  
  end; { DoUpdate }  
  
procedure DoDisk;((MountDisk))  
{ ; Handle disk-insertion  
{ ; mounted properly  
  
var
```

LightsBug

SRC Zone: 0105C2

registers	zones
DoUpdate	H 010B1A 001867 2 PICT 0080 108
MainLoop	H 010B16 00185C 2 PICT 0081 108
Animator	H 010B12 001839 2 PICT 0082 108
	H 010B0E 00182A 2 PICT 0083 108
	H 010B0A 00001A 2 STR 0080 108
	H 0106FA 0001A2 0
	H 010ADE 000048 2 MENU 0003 108
	H 010ADA 000066 0
	H 010AE2 000031 2 MENU 0002 108
	H 010AE6 000181 2 MENU 0001 108

Base: 00000000 + Offset: 0000 Edit

000:	00F8 0000 FFFF FFFF 000F 69F6	..0..0000..i0
00C:	0000 CFF0 0000 CFF2 0000 CFF4	..e0..e0..e0
018:	0000 CFF6 0000 CFF8 0040 1134	..e0..e0.@.4
024:	000F 67D6 000F 75E8 0000 CFFA	..g+..u0..e0

UEvent handles all the individual events. **DoMouseDown** asks where the mouse was clicked. If it was clicked in the menu, **DoMenu** finds out where and opens a desk accessory, displays the About Animator... alert, or sets the Backwards or Quitting flags. It also sends **Edit** menu commands to desk accessories for processing. If the mouse was clicked somewhere else, **DoMouseDown** appropriately selects or drags a window, passes the click onto a desk accessory, or ignores the click.

DoKeydown is simple in *Animator* because it only has to process menu equivalents chosen with the command key. The **MenuSelect** call processes command-key equivalents for us. Activate events are processed by **DoActivate**. Besides just bringing selected windows to the front of the screen and highlighting them, **DoActivate** enables and disables the **Edit** menu so it is unavailable to the application but is available for desk accessories. **DoUpdate** is the routine that refreshes the application window on every pass through the main event loop and really does the drawing. Lastly, **DoDisk** takes care of mounting disks and allowing the user to initialize unformatted disks.

Left as an Exercise for the Reader

You can customize *Animator* by creating any 'PICT' resources (using MacPaint, MacDraw, etc., and copying the images into the Scrapbook) and pasting those 'PICT's into *Animator* to replace the ones already there. Start numbering the 'PICT's with resource IDs of 128 and up (sequentially). Each 'PICT' should be the same size, but you can add as many 'PICT's as you can fit. Each 'PICT', from FirstPict (currently 128) up will be displayed in turn.

BMUG
Fall 1986



Animator is fairly robust. I'm sure I've violated a rule here and there and overlooked a few things, but *Animator* checks for a few simple errors. If a desk accessory does not load (maybe because enough memory is not available), *Animator* displays an alert to let the user know. If no 'PICT's are available, *Animator* just stops and quits. If the available 'PICT' resources lead to a window that is bigger than available screen space, the window is made as large as possible and only some of the 'PICT' is shown. Despite these checks, I have undoubtedly left some stones unturned. I have compiled *Animator* with an 8K stack and a 128K heap, and loading in lots of large 'PICT's will certainly lead to an out-of-memory error. As an exercise to see if you're awake, you might want to add a few lines of memory checking to preview how many pictures are to be drawn, how large each is, and how much space is available. If you find that not enough space exists, display the (as yet unused) ID=204 alert and **ExitToShell**. Or you might want to get fancy and ask the user if they mind displaying just as many 'PICT's as will fit, or maybe try growing the heap zone to accomodate the 'PICT's. The possibilities are almost limitless.

Linda Custer is a Ph.D. candidate in Berkeley's Chemical Engineering department and a freelance Mac writer, programmer, and consultant.

Msg. #3948 in **Public Messages** Posted on 10/20/86 at 22:00:37
To: ALL From: GEORGE LANGDON
Subject: Pascals for the Mac
I concur with Claude that LightSpeed Pascal is a good chioce for both learning Pascal on the Mac and obtaining a useful, compiled language. I have been at it for about three weeks. LS is really the same as MacPascal from Apple...actually both where written by Think. The syntax is the same, but with LS you can write real programs. The price is right as well. I paid \$125 at Winners Circle, have seen it for \$100 at Software for Less, and in MacTutor there is an ad listing a price of \$77. After reading through Moll and a couple for other standard Pascal books, I got frustrated with their limited info on programming for the Mac. I'm now working through Christopher Morgan's book 'Hidden Powers of the Macintosh' from Plume. Well written and actually easy to understand if you work through the programs in the book. Most can be entered directly in LS Pascal, but beware...some have outright bugs that should have been caught in editing. Hopes this helps and responds to a couple of message at once. GO FOR IT!!

BMUG
FALL 1986

Msg. #3144 in **Public Messages** Posted on 09/11/86 at 00:08:31

To: ALL From: VANCE BRADFORD

Subject: Lightspeed Pascal

I recently purchased Think Technology Pascal, and wow!!!! It has a fantastic symbolic debugger, just like the original Macintosh Pascal, and it's really, really fast! It also has the best user's guide and reference manual I've seen for a compiled development language. I am really impressed! Think Technology has done it again.

One mans opinion,

Vance Bradford, Murray, Ut.

Msg. #3335 in **Public Messages** Posted on 09/18/86 at 20:13:05

To: ALL From: GEORGE LANGDON

Subject: LightSpeed Pascal

Well, I may be the first on my block to get a copy of Lightspeed Pascal. I happened upon a copy at Winners Circle weekend before last. After two weeks I can only say that it appears quite solid, very easy to use, and basically a compiler version of Macintosh Pascal. I am no Pascal programmer, and so have been using Hayden's book "Intro to Macintosh Pascal" to learn. The programs and instructions for using the language are identical to LightSpeed. LS does have a very slick source code debugger; however, I haven't gotten into any jams to use it, yet. I have found that longer programs cannot be entered as one source file. LS has a limit of 32,000K on file size; so you must break-up longer programs into Units. Not really difficult, just another thing to learn. The LS manual does not teach you much about Pascal. Anyone have good suggestions on primers on Macintosh Pascal??

If you have questions, just ask...

Msg. #3483 in **Public Messages** Posted on 09/26/86 at 15:34:09

To: ERIK TIEMROTH From: RAINES COHEN

Subject: Reply To 'pascal compilers'

Erik -

I like LightSpeed Pascal a lot, as it is blindingly fast and a terrific debugging environment. It has a couple of bugs but they are usually work-around-able. I saw Turbo Pascal for the Mac yesterday at the PC Faire in San Francisco, and it looked VERY fast but it doesn't offer the kind of debugging that LightSpeed does. TML, with version 2.0, supports objects (like MacApp and MPW Pasccal), and now also runs faster.

-- Raines Cohen

Co-founder, BMUG Beginning Developers SIG

(meets every other Wednesday in Wheeler hall) (call 849-9114)



ZBasic: A Quick Review by a BASIC User

Ken Huie

Like the ad says

ZBasic is a BASIC compiler that is able to produce stand-alone applications and maintain the interactive (quick and dirty) nature of BASIC. It also includes many of the extensions of Microsoft (MS) *Basic* useful for programming the Mac interface, such as the MENU, ON MOUSE, etc. commands, although they are implemented somewhat differently. The disk is sold with the *ZBasic* application (100K), a help file (45K), *Edit* (61K), *RMaker* (11K), and the standard System/Finder.

Outwardly, the *ZBasic* program is clean and well organized; however, I was unable to use it without first reading (the first Mac test) the manual, a 500+ page softbound book. *ZBasic* has a "configure" selection on the menu bar to change the type of default variable (integer, long integer, single precision, double precision) and set the number of BCD digits. Variables default to integer form and I was forced to start using type declarations (\$,!,#) at the end of variables' names (goodbye quick and dirty). By the way, calculations with real numbers using BCD math are slower than integers but numbers with up to 240 digits (with no round-off) can be controlled. Cryptographers will love this. Labels are supported but they must be enclosed in quote marks, one of several differences from MS BASIC.

Noticeably missing is the ability to declare arrays using a variable. All arrays must be declared with an explicit number. [ie: DIM AnArray(25) vs. DIM AnArray(AVariable)] Although *ZBASIC* advertises portability between the Macintosh, IBM PC, Z-80, and Apple // versions of *ZBASIC*, this portability is more theoretical than real for those wanting to implement the Mac interface, or utilize features such as serial port or MacinTalk access.

The editors

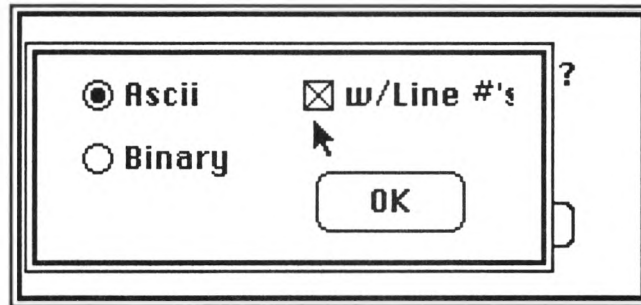
ZBasic comes with *three* editors. Two editors operate within *ZBasic* and one is the familiar application *Edit*, which has a lot of features, fonts and the ability to show invisible characters.

ZBasic's command window contains the second text editor: a traditional line editor. It operates quite like a scrolling ASCII terminal with a command structure similar to *MBASIC* on a CP/M machine. This is expected since *ZBasic* is available on Z-80, Apple II, and MS-DOS machines. Listings within this window have auto-indenting of loops. The **Edit** and **List** commands are the only way to view text in this window. When you run a program without specifying an output window, the output appears in this default window.

The third editor, toggled in/out with Command-E, is a Mac-type text editor that allows mouse movement and scroll elevators. This window editor is sometimes faster than the line editor but you cannot RUN from this window; you have to toggle back to the command window using Command-E. In early versions of *ZBASIC* this editor barely worked at all; as of version 3.01 it still exhibits idiosyncracies such as the occasional misplacement of lines.

When entering lines inside the command window, *ZBasic* needs a unique line number on each new line of text. The auto-indent feature is not present in the text editor window. Although *ZBasic* has an auto-line-number function and existing text files (like from *Microsoft BASIC*) receive line numbers when you import them, you can only remove line numbers from existing files by utilizing an undocumented feature.

To save without line numbers, click on the close box of the command window after making a change to your program. ZBASIC will then respond with the dialog box "Save changes before closing?" Click on the **YES** button. This will bring up the following dialog box:



Click off the **w/Line #s** button, and **OK**, and you're home free.

No simple "search and replace" function is available within the two ZBasic editors. This is annoying, since ZBasic is case sensitive to variable names and finding a variable name with an incorrect uppercase letter can prove to be frustrating. A **Find** command, provided as a substitute for "search and replace", is very awkward to use. One easy way to get around this is to install a text editor desk accessory such as MockWrite that implements **Find**.

Documentation

The book has around 500+ pages, with about 100 pages devoted to MSDOS, CP/M, and Apple II specific commands and considerations. The section on Mac specific commands is about 115 pages. About 20 pages discuss how to convert from other BASICs, and includes a convenient list of commands *not* available in ZBasic and what command to use instead. The remaining 400 pages cover commands and graphics common to all versions. Lots of multi-line examples are scattered about. The book has a 6 page index that helps find most needed topics. MacInTalk, AppleTalk, and serial port access are present as special functions and the toolbox is minimally supported, but you need *Inside Macintosh* to sort out the procedure calls. The book simply warns that you "use them at your own risk" (ugly).

Mac Interface Support

The easiest way to program the Mac interface in ZBASIC is to utilize the built in commands for creating windows, implementing menu choices, scroll bars, help windows, and edit fields. Alert windows, BREAK, DIALOG, MENU, MOUSE, and TIMER event trapping, BUTTON support, FILES\$ (which brings up the familiar file saving or selection dialog boxes), and laserwriter support are all implemented, although you'll find yourself having to write example programs to figure out exactly how they work. Serial port and MacInTalk access is provided via the OPEN statement. To get serial port access (or statements such as INKEY\$) to work correctly, remember to turn off the BREAK, DIALOG, MENU, MOUSE, and TIMER interrupts.

ZBASIC supports in-line inclusion and execution of machine language code, via the CALL statement. Users programming the Mac interface in ZBASIC will outgrow the documentation within a short period of time. One possible source of information is the ZBASIC section within the Macintosh developers group on GENIE.

Some results

Typing "RUN" error-checks and compiles the program. Since the Sieve program benchmark is published in the ads for ZBasic, I chose to try some programs already written in Microsoft

BMUG Fall 1986



BASIC not so much to see how fast they run but to test the compile time and ease of conversion to a stand alone application.

The first program "Hello" in *ZBasic* is equivalent to this familiar program in Pascal.

10 PRINT "Hello"	PROGRAM (output);
20 END	BEGIN
	WRITELN('Hello')
	END.

The second program is from a recent *Byte* article (Aug 86), "Palindrome", that uses long precision BCD math and lots of *STRING\$*. This program *RUNs* like a jet-set bandit.

<u>Program</u>	<u>Compile time</u>	<u>source size (bytes)</u>	<u>code size (bytes)</u>
"Hello"	39 sec.	19	34342
"Palindrome"	35 sec.	1713	36720

Note that these times are the times needed to compile to a 400K disk and possibly varies with the most recent disk operation; 800K disks are a bit faster. *RUNing* a program without compilation to disk is a more practical way to use *ZBasic*. Compiling destroys the ability to interact and debug a program. Test any program *a lot* before you bother to compile it. The size of the code generated while building the application demonstrates the lack of an intelligent linker; the whole library is added to each application. An equivalent program (Hello) in Pascal compiles to an application of less than 4K.

Conversion from Microsoft *BASIC* to *ZBasic* can be tedious and is non-trivial; although an MS *Basic* to *ZBASIC* translator is available, it does not handle all aspects of the conversion. For example, *ZBASIC* implements event trapping differently, requires quotes around statement labels, uses numbers rather than labels in a *RESTORE* statement, and has different syntax for *IF-THEN-ELSE* constructs. The default variable type has to be selected and the type declaration has to be added to all variables not of the default type. Upper and lower case spellings have to be matched. Re-numbering of lines is automatic but groups of lines are not easily moved; manual re-numbering is needed—a royal pain in a large program. The *Edit* application helps some, but the only test is to *RUN* the program and wait for crashes (some fatal, with bomb). In fact, since the programs are compiled when *RUN*, some *simple* errors are fatal if error trapping is not turned on. Turning on the *BREAK* function—so you can halt the program with command-period—slows the operation of a program noticeably, and will disrupt serial port access and keyboard reading via the *INKEY\$* function.

Value

At about \$90 (around \$50 for mailorder) *ZBasic* is a moderate value. The documentation is readable and accurate. The applications produced have a minimum size of about 34K. The saving graces of *ZBasic* are its speed (on the order of available C compilers), the interactive nature of *BASIC*, and the ability to produce stand alone applications that implement the Mac interface without learning the Toolbox calls. Version 3.01 does not fully support HFS, but version 3.02 should. Learn another language if you don't already know *BASIC*.

ZBasic, \$89.95
Zedcor, Inc
4500 E. Speedway, # 93
Tucson, Arizona 85712

Bypassing The Trap Dispatcher For Fun And Profit

James E. Hopper

Car 54 Where Are You?

One of the secrets of the Apple II success story was the support provided to the hackers. Apple actually published the source code for the ROM's in the reference manual (the one that replaced the early Red Book). With this wonder of documentation, a programmer could dramatically cut down the work in writing an application by calling the ROM routines (kind of familiar isn't it). However like most blessings it was not unmixed, and Apple is still paying for documenting these entry points as physical addresses in the ROM. This has forced them to maintain each of these entry points at the same address with each revision. Thus if a ROM procedure gets longer, there is usually no alternative but to put the end of the procedure off somewhere in the ROM and have the beginning portion jump to it. This has tended to turn the Apple II ROMs into a maze of patches, and made it very difficult for the ROM to be upgraded.

With the Macintosh, Apple had a second chance. What they did was to have the ROM install a dispatcher in RAM at boot time, and all access to the ROM is intended to be through this procedure. Thus it is possible to patch or change the address of a procedure in ROM by simply changing the address in the RAM resident Trap Dispatch Table. When the ROMs come out, the procedure addresses in ROM are moved about. Thus, if everyone follows the guidelines in writing their applications, they will all work without having to change the software with every new ROM release. The price paid for this is that every call to a ROM procedure has some overhead added to it as the dispatch procedure is called whenever a TRAP is executed. Thus, an extra call is made for every ROM command encountered. While normally this overhead is very small, if a trap is called enough times, it can add up and as a result slow down the execution of the program. This is especially true for those procedures whose execution time is comparable to the time to call them through the Trap Dispatcher.

Where Is That Bypass?

Some programmers have been known to speed up their code by calling ROM procedures directly, bypassing the trap dispatch procedure. This can be a very bad practice if done incorrectly, that is, by hard coding the ROM address into the code. While the resulting program will work, the first time the ROM is changed the program may crash and burn! One should note that it doesn't take a ROM revision to change a procedure's address, the 'INIT' and 'PTCH' resources both change the Trap Dispatch Table addresses for ROM procedures every time you boot your system. This is done to patch bugs or add features to existing ROM procedures. However if your program is calling the procedure in the ROM it is likely bypassing these fixes. A correct way to bypass the trap dispatch procedure is suggested in Scott Knaster's book, and is just as fast as the wrong way. Call the trap **GetTrapAddress** once in your program and get the current address of the trap procedure you are interested in. As Scott says, Apple will guarantee that the ROM procedure's address will not change while you are running your program.

Things Can Get A Bit Sticky!

To use this technique in a higher level language, like Pascal, you need what is called a "glue routine". What this Glue routine will do is correctly set up the arguments and call the ROM procedure at the address you got from the **GetTrapAddress** call. For example, in TML Pascal the call to the ROM procedure **BitTst** is defined as:

```
Test:= BitTst (bytePtr,bitNum);
```



The compiler will place the proper arguments (bytePtr, bitNum) on the stack, then insert the Hex \$A85D in the object code. The use of the **inline** directive obviates the need for a glue routine in this case. The code generated by the compiler follows:

```
;;      Test := BitTst (@bytePtr, 1);
      clr.b          -(SP)
      lea            bytePtr(A5), A4
      move.l         A4, -(SP)
      move.l         #1, -(SP)
      dc.w           $A85D
      move.b         (SP)+, D6
      move.b         D6, test(A5)
```

Space is reserved on the stack for the Functions result of **BitTst**. A pointer to the first byte in the block is pushed onto the stack. A "1" (for first byte in the block) is pushed on the stack. It should be noted that using the actual numbers in a program is frowned upon. However, this is a hack, not a real program. The 2 bytes of the trap address are then put into the code (that's what **inline** really means). This will cause the processor to trap (or intercept) the instruction and jump to the trap dispatch procedure. The dispatcher returns control to the next statement after the trap is executed. This statement pops the function result off the stack, then moves it into the Pascal variable Test. This scheme must be slightly modified to bypass the Trap Dispatch Table, and will require the use of a Glue routine. Our Glue routine, which will replace the standard **INLINE**, will need the address of the trap procedure as an additional argument. It must clean up the stack to insure the arguments are correct. Then and only then will it do a jump to the ROM procedure. The call is as follows:

```
Test := NTBitTst(bytePtr, bitNum, BitTstTrapPtr)
```

The code generated by the compiler is as follows:

```
;;      Test := NTBitTst(bytePtr, 1, BitTstTrapPtr);
      clr.b          -(SP)
      lea            bytePtr(A5), A4
      move.l         A4, -(SP)
      move.l         #1, -(SP)
      move.l         bittsttrapptr(A5), -(SP)
      jsr.w          NTbittst
      move.b         (SP)+, D6
      move.b         D6, test(A5)
```

The differences here are small, but significant. First, one more argument is pushed on the stack, BitTstTrapPtr. Please note that at this point, the stack is identical to the stack for the dispatcher based call, except for this last argument on the stack. The next instruction is a JSR to our glue routine, **NTBitTst**, instead of the **inline** trap address, which will push the return address onto the stack. The end of this routine is the same as before.

Listing 1 is the assembler source for a **BitTst** glue procedure which bypasses the Trap Dispatch Table. The Pascal compiler code copies the three arguments to the stack, as we saw above. The return address is then pushed on the stack and control is transferred to the glue procedure. The glue procedure then pulls the trap address off the stack, pops the return address off the stack,



and puts it back on the stack where the trap address was (which is where the trap procedure expects it). While the stack manipulation looks a little strange, if you read your processor guide you will find that it is the most efficient way (most people when they see this for the first time tend to believe the compiler screwed up).

Listing 2 is a small program that I wrote to time how long the **BitTst** trap takes, calling it via the Dispatch procedure verses calling it directly. The sole purpose is to find out what the time difference is between the two methods of calling **BitTst**. Since all the code is identical except the calls to **BitTst**, the delta time is the time saved by bypassing the Trap Dispatcher.

Cruising 675!

When this test procedure is run, the time on my Mac+ for the dispatcher version is 54 ticks, while the time for the non dispatcher version was 38 ticks, or approximately 70% of the dispatch version time, note that the absolute times here are quite small. Thus this technique, as I mentioned at the start of this article, is useful only if you are going to call the trap a lot of times. What you might ask would call a trap procedure this many times? Well, for instance, a recent program John Matthews and I wrote converts an Apple II HiRes screen into a Mac bit map. This procedure calls **BitTst** 51,570 times for each picture, and **BitSet** for as many of these times as the Apple II HiRes bit is on.

This technique used for both **BitTst** and **BitSet** cuts slightly over 1 second out of the conversion time of 8 seconds using the dispatch table. While this is not a huge difference every little bit counts and only required about 10 lines of assembly code. Those lines can be added to my library of standard procedures for later reuse. Ok, I'll admit it makes little to no difference in this application but it was such a neat hack I couldn't resist. Besides it would have saved a larger amount of time on a full page graphic, or a 300 dpi bitmap for use with the Laserwriter. Let's see, an 8x10 bitmap at 300 dpi is 7,200,000 bits. If it called this routine that many times theoretically would save 192 seconds. Now that's more like it. I suspect this technique is only going to be of interest to those folks trying to squeeze the last little bit of performance out of their application, or perhaps image processing where you have a lot of bits to twiddle.

From Apple-Dayton Journal (Oct 1986), the newsletter of the Apple Dayton Users Group, Dayton, OH.

Jim Hopper is a long time Apple computer enthusiast, who bought his first Apple back in 1979. He has written articles for Call A.P.P.L.E. and Apple Orchard, in addition to the Apple Dayton Journal. He has written several PD utilities for both the Apple II and the Macintosh, including DiskScan, FastBoot, DiskMap, Sleep Key, DiskScan, and AppleII Mac. He also has the distinction of writing the first arcade game for monkeys! He is does not however have his own personal picture icon like all the big name authors, yet (grin).

Listing 1

```
XDef NoTrapBitTst
;Function NTBitTst(BytePtr:Ptr;BitNumber:LongInt;TrapAddress:Ptr):Boolean;
    NTBitTst
; the stack has one too many arguments for this trap
; procedure so we must take the address off the stack and move
; the return address of our calling program in its place in order to
; insure the trap procedure sees the stack the same way it
; would if we had called the Trap Dispatcher
; get the address of the trap procedure from stack
```



```

        Move.L          4(SP),A0
; restore the stack to the way the procedure expects it
        Move.L          (SP)+,(SP)
; jump to the ROM procedure (which will return to the caller)
        JMP             (A0)

```

Listing 2

```

program TestDispatcherTime (Input, Output);
{$T APPL JIMH      }
{$I Tecmar:SoftwareDevelopment:PascalSystem:MemTypes.ipas  }
{$I Tecmar:SoftwareDevelopment:PascalSystem:QuickDraw.ipas }
{$I Tecmar:SoftwareDevelopment:PascalSystem:OSIntf.ipas    }
{$I Tecmar:SoftwareDevelopment:PascalSystem:ToolIntf.ipas  }
{$U Tecmar:SoftwareDevelopment:WorkingFiles:BitTst  }
{$a+}

var
    DummyAddress      : Integer;
    i                  : Integer;
    StartTime          : LongInt;
    EndTime            : LongInt;
    DispatchDeltaTime  : LongInt;
    NoDispatchDeltaTime : LongInt;
    BitTstTrapPtr      : Ptr;
    DispatchTimeString : Str255;
    NoDispatchTimeString : Str255;
    Test
    : Boolean;

function NTBitTst (BytePtr : Ptr; BitNumber : LongInt; TrapAddress : Ptr):
    Boolean; External;

begin
    { get the current address of the BitTst routine }

    BitTstTrapPtr := Pointer(GetTrapAddress($5D));
    StartTime := TickCount; { start the clock }

    { loop 10,000 times calling the function BitTst via the dispatch routine }
    { to get a fair sample }

    for i := 1 to 10000 do
        Test := BitTst(@DummyAddress, 1);

    EndTime := TickCount; { stop the clock }
    DispatchDeltaTime := EndTime - StartTime { compute the time }
    StartTime := TickCount; { restart the clock }

    { loop 10,000 times calling the function BitTst directly }
    { to get a fair sample }

    for i := 1 to 10000 do
        Test := NTBitTst(@DummyAddress, 1, BitTstTrapPtr);

```



BMUG FALL 1986

```
EndTime := TickCount; { stop the clock }
NoDispatchDeltaTime := EndTime - StartTime; { compute the time }

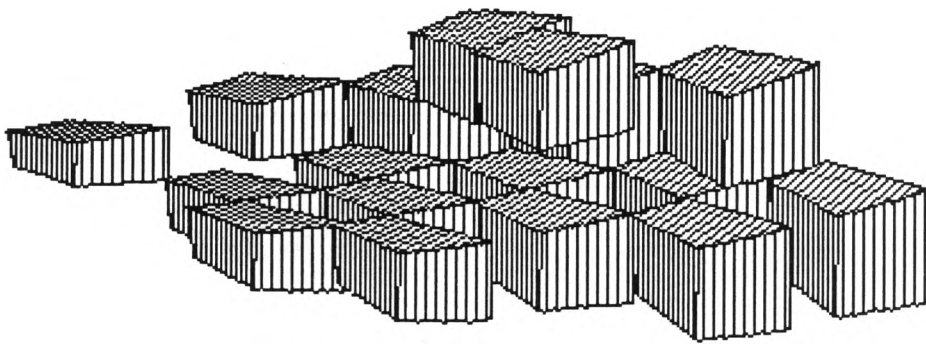
{ convert the times to strings for output }

NumToString(DispatchDeltaTime, DispatchTimeString);
NumToString(NoDispatchDeltaTime, NoDispatchTimeString);

{ print out the two times for the user }

Writeln('Dispatch time for 10000 iterations of BitTst: ',
    DispatchTimeString);
Writeln('No Dispatch time for 10000 iterations of BitTst: ',
    NoDispatchTimeString);
ReadLn; { let user read the results }

end.
```



Boxes/Hockabout

BMUG
Fall 1986



*Productivity
Tools*

PRODUCTIVITY TOOLS

File Finders

- Three programs that locate files

Mark Loncar 93

Koala KAT-ing Around the Classroom

- A mouse alternative

Kenneth William Hirsch 96

Micrograde and the Joys of Teaching Classes

- Grade bookkeeping and calculations

Jasper Rine 98

All the Home Accounting Programs

- Five programs reviewed

Scott Bearer 101

Mini Reviews

- On time, tax, and disk zapping

Scott Bearer 107

Smart Alarms

- An appointment DA

Steve Cisler 110



File Finders

Mark Loncar

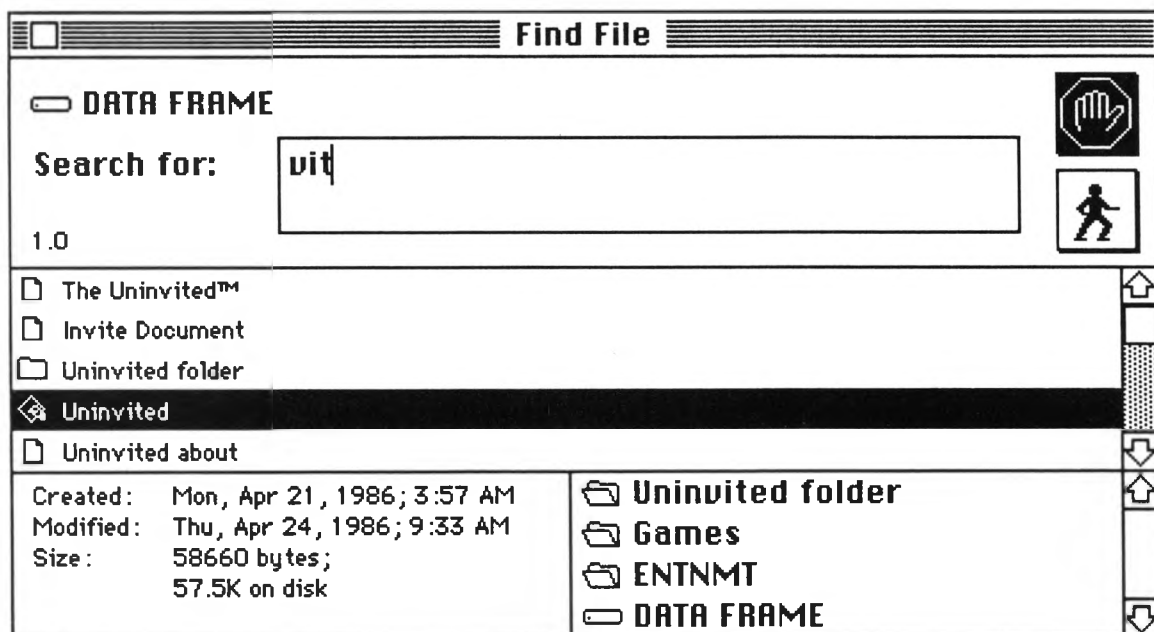
The Problem

As the prices of 20-megabyte hard disk drives drop, more and more Macintosh users are buying these mass storage devices to deal with their growing libraries of documents and applications. An unfortunate side effect of mass storage is the ease with which files that are hidden away in some forgotten folder can be misplaced or "lost" in the new Hierarchical File Structure (HFS).

Unlike the old Macintosh File Structure (MFS), HFS can only find files within the folder selected. If you have a document in a folder that is nested three or four generations deep in other folders, it may be hard to find, even if your system is generally well-organized. Luckily, there are Desk Accessories that can find a desired file, without having to leave the current application, by searching all folders for you. This article describes three of these filefinding DA's: Apple's *Find File*, Dave Dunham's *Disk Info* (v 1.43 or greater), and PBI Software's *HFS Locator Plus*. None costs more than \$40.00, and two are less expensive still.

Find File

Apple's *Find File* is the newest and least expensive. Released with Apple's new 20SC SCSI hard disk drive in September, 1986, and installed with the Font/DA mover, it takes up 11.5k on your disk. Selecting it from the Apple menu displays the window shown below. Type in the name or part of the name of the file you wish to find and click the Proceed icon (the button with the little man in it) or hit Return. *Find File* will search all your folders and display all file names containing your search characters. Preceding each file name is an icon (similar to those in the Finder when viewing by size) describing what that file is (document or application). Click on the file to "get info" on the file (size, creator and type) in one window and see its hierarchy of folders in another window.





BMUG FALL 1986

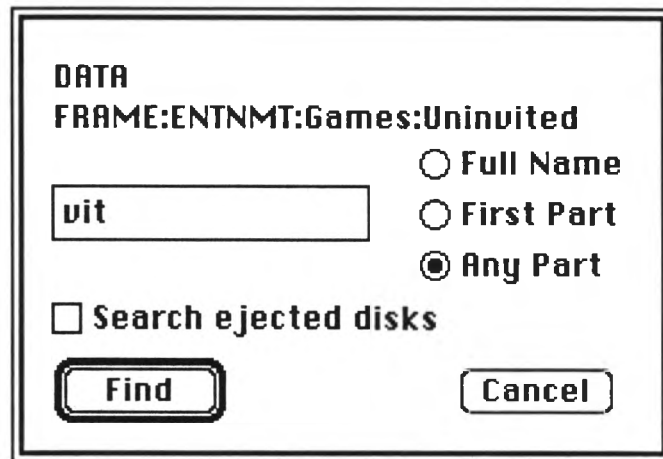
Find File can run in the background of another program in a multitasking fashion. Give it search instructions, return to your current program, and *Find File* will continue to search for your request, beeping when complete. This drastically reduces the speed of your current application, but can be useful in many situations. If you don't want to complete a search, simply click the "stop sign" icon to abort. You can also begin a search at a given location or folder to shorten search time.

Find File is free to purchasers of the new 20SC. The rest of us may have access to it in a fashion similar to the System and Finder updates, but distribution plans aren't settled yet.

Disk Info (v. 1.43)

Disk Info, by Maitreya Design, is the oldest of the three programs, having been around nearly as long as the Mac. The program was written by Dave Dunham, who wrote the Desk Accessory outliner Acta. Although *Disk Info* has been around a while, it only recently acquired the ability to find files.

The "Find" function of *Disk Info* is rather sparse, but it does work. Unlike *Find File*, *Disk Info* has no icons in its dialog box and can't work in the background, but searches for files based on the "Full Name," "First Part" or "Any Part" of the file name. *Disk Info* is faster than *Find File* by a few seconds, but cannot abort a search.



Disk Info's "Find" Window

Disk Info is not primarily a file finder, but a hard disk management tool that allows the user to delete, rename, and get info (size, creator, type) regarding a file, folder, or disk. It also can compact the Random Access Memory, sort files and folders, and view file size in characters or "k" of memory. *Disk Info* does not have the best filefinding capabilities, but compensates with other virtues, and all for \$10.00.

HFS Locator Plus

HFS Locator Plus is probably the most advanced and fastest file finder of the three, using excellent graphics and taking full advantage of the Macintosh interface.

HFS Locator Plus allows you to create folders, launch and move applications, and rename, copy, and delete files. Click on the Show Path icon, and a menu drops down showing the



hierarchical path to the file. You can also create a text file catalog and put all the findings from a search into that file. Besides searching by name, you can search by date of creation or modification, find files for the current application, find files for a different application, search for invisible files, and the program provides you with helpful hints.

One drawback with *HFS Locator* is that it cannot search for any part of a file name the way *Find File* and *Disk Info* do. *HFS Locator* will only search based on "Begins with" or "Matches" (not "Includes" a specified string). It is otherwise a very nice locator with the ability to launch a totally different application at the close of the current application (which is faster than going through the Finder).

File "MacWrite 4.5 " is in "WORD PRCS".


Show path

Cancel

Close

Dispositions

Rename	Move
Delete	Copy
Show info	Launch

Create folder

Catalog

Find next

Locator Search Status

% Searched: Matches found: 1

Suspend search

HFS Locator Plus

Conclusion

Each of these programs is useful and inexpensive, and each is unique in its abilities. If you're need a simple locator that works in the background, get *Find File*; if you need an inexpensive Desk Accessory that has more features than a simple locator, you can't beat *Disk Info* at \$10.00. *HFS Locator* is the most comprehensive of the three and has some nice features, for which you pay more. Any will find a given file, and all three work. If you're lost in the catacombs of your hard disk, put one of these to work and find your way out.

Find File

Apple Computer, Inc.

Disk Info (v. 1.43)

Maitreya Design - ShareWare
BMUG Disk #11

HFS Locator Plus

PBI Software

Koala KAT-ing Around the Classroom

Kenneth William Hirsch

If used as part of a Macintosh classroom teaching system for in-class demonstrations, installed on a roll-around audio-visual cart, the mouse presents problems:

- (1) Lecturers frequently must move away from the cart to write on the blackboard or to check their notes. While away from the cart, it is not possible to use the mouse; the lecturer must defer manipulation of the computer until he returns to the working surface of the cart top.
- (2) With the Macintosh computer, its cables and disc drive placed on top, most carts become cluttered; mouse movement becomes circumscribed and subject to interference.

The Koala "KAT" alternative to the mouse, a stylus or finger-operated pressure tablet, seemed likely to offer a solution, as:

- (1) it would be possible to carry the KAT to a black board or other position and, when needed, move the cursor or Mac screen image around and initiate commands to illustrate the presentation.
- (2) when on the cart, the KAT would do well at defending its own territory against clutter.

After a month-long evaluation of the KAT for such use, I conclude:

- (1) The KAT remains the only convenient cheap, hand-holdable, device capable of positioning the cursor and commanding the Macintosh. As such, it would work in place of the mouse to enable use of the Mac mounted on a cart while lecturing in a classroom.
- (2) The KAT, as currently made, presents some problems:

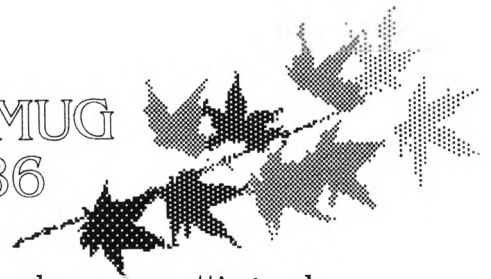
a. Use of the finger or stylus in its quick, direct mode results in variable, jagged lines and cursor placement.

Clicking on the smoothing function reduces the variability to make the KAT usable, but in doing so, introduces lag time which results in cursor overshoot; the user moves her finger across the plastic surface of the KAT and stops where the cursor should stop -- but the cursor goes a bit too far.

It seems likely that the smoothing function introduces a delay into the circuit which acts to draw on-screen the changes in stylus/finger position as they occur on the KAT surface. The delay would be coupled with a moving average of a variable number of stylus/finger position data points, to "smooth" the cursor on-screen. The problem is that the more the smoothing, the larger the number of data points averaged and the greater the "inertia" of the stylus movement. Koala needs to create a more sophisticated design.

b. Stylus use is O.K. only when movement is crisp. When movement is minute and slow, as is required in detailed drawing, the KAT "skips" a great deal.

It is probable that more experimentation with settings of sensitivity and smoothing could improve the situation, but each user would have to spend time with the KAT to find her own best



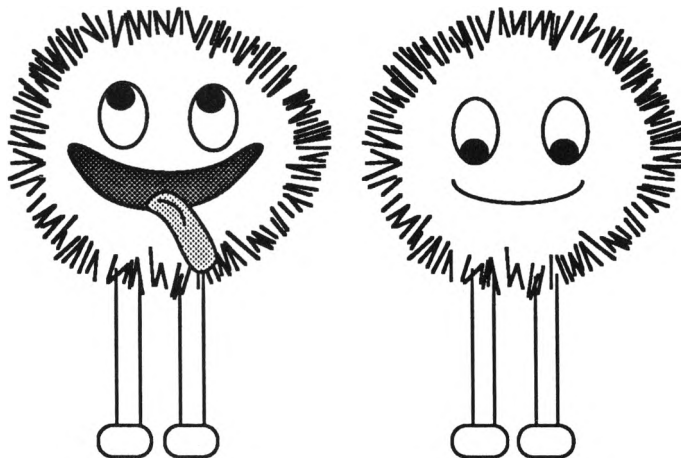
settings, thus invalidating KAT usage with stylus in participatory classroom settings where there is no extended opportunity for each user to make trials and adjustments.

c. the switching function which is the equivalent of mouse "drag" is inconvenient.

Nevertheless, despite the problems mentioned, KAT is currently the best way of dealing with the lack of mousing surface associated with using a portable, cart-mounted Mac system to accompany classroom lecture/student interaction with the Mac computer system.

The "KAT" system is available at a list price of \$199, but can be obtained from typical discount supply houses for \$135 plus shipping. Information may be obtained from:

Koala Technologies
2065 Junction Avenue
San Jose
California 95131
phone: 1(800) 562-2327



Puffs/Dan Wood

Micrograde and the Joys of Teaching Undergraduate Classes

Jasper Rine

Of the many joys of teaching undergraduate classes at a large university, the bookkeeping involved in grading is not one of them. Even in the spoiled and pampered position of having graduate teaching assistants who readily and cheerfully grade exams, and enter and compile the data, I still find that the basic job of maintaining grade records excessively time consuming, utterly boring, and since it is boring, prone to errors. These errors tend to be the clerical type that are difficult to track down, and capable of causing high anxiety among the students. It was inevitable, at a time when there are Macintosh programs for tasks as mundane as balancing a checkbook, that someone would write a program to handle grade bookkeeping and calculations. *Micrograde*, by Chariot Software of San Diego, is a recent arrival in this field. The acid tests that need to be applied to all such "work-saving" programs are (1) Does the program take more work than the way you were doing the job before?; (2) How badly can the data be screwed up by bleary-eyed teaching assistants or professors?; and (3) Are the results worth the effort to anyone other than a Macaddict? In my opinion, *Micrograde* comes close to filling the bill. Before I discuss its virtues and few limitations, I will describe the teaching situation that I deal with in order for you to understand the prejudices in my evaluation.

I team teach a large undergraduate class that typically has 150 to 175 students enrolled. The grade each semester is based upon the cumulative score on three hourly exams and a final exam. Presently, we use programs run on a VAX-Unix operating system to record and process grade information. Thus, each semester a teaching assistant volunteer has to manually enter student names, student numbers, and test scores into this program according to the rights and rituals of Unix. We have a resident guru who patiently explains the Unix voodoo to each years' teaching assistants. Unfortunately, our guru will soon earn his degree and leave us to either adopt new software or resort to our pocket calculators. *Micrograde* passes the first test easily, since entering the data in *Micrograde* is no worse than Unix and it is much easier to edit errors in *Micrograde*. One slight complaint that I have is that it is possible to enter the same name twice without raising the program's eyebrows. This type of error could lead to significant clerical problems later if it isn't corrected.

The data and calculations appear in four tiled windows that present different types of information about the class. One window is a **Class Roster** that can list the students by any criterion that one wants. The second window is the **Student Summary** window. When a student is selected from the roster window, the data concerning that student's entire performance in the class is presented in this window. The third window is the **Class Scores** window. This window contains a class roster where the performance of the entire class can be displayed for any assignment by simply scrolling through the possible assignments with the scroll bars at the top of the window. The fourth window is called **Categories and Assignments** and it is in this window that the grading structure of the course is defined by the instructor in response to simple menu formats. The visual effect of the windows is very clear and intuitively organized and represents all aspects of classroom bookkeeping. The windows exploit the Mac's visual approach to program operation and should be self-evident to anyone who has any experience with other Mac software.



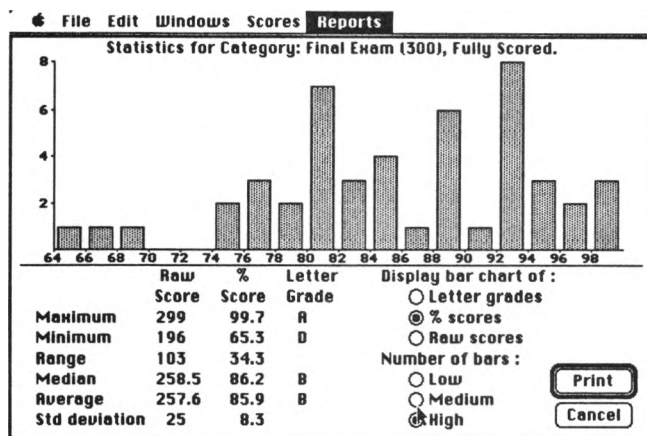
File Edit Windows Scores Reports													
☐ Roster for test db													
Class has 44 students. Total points: 68													
☒ Name Hambrecht, Betsy Heckendorn, John Huxham, Fred James, Jay Jones, Reese Kane, Beverley King, Tony Kowalski, Nicole Kronick, Scott	Student Summary Jones, Reese I.D. 72729. Grade Average: 558, 82												
☐ Categories & Assignments Contains 5 categories, 5 assignments Midterm1 (100) Midterm2 (100) Midterm3 (100) Notebook (80) Final Exam (300)	<table border="1"> <thead> <tr> <th>Category</th> <th>Assignment</th> </tr> </thead> <tbody> <tr> <td>Midterm1 (100)</td> <td></td> </tr> <tr> <td>Midterm2 (100)</td> <td></td> </tr> <tr> <td>Midterm3 (100)</td> <td></td> </tr> <tr> <td>Notebook (80)</td> <td></td> </tr> <tr> <td>Final Exam (300)</td> <td></td> </tr> </tbody> </table>	Category	Assignment	Midterm1 (100)		Midterm2 (100)		Midterm3 (100)		Notebook (80)		Final Exam (300)	
Category	Assignment												
Midterm1 (100)													
Midterm2 (100)													
Midterm3 (100)													
Notebook (80)													
Final Exam (300)													
Scores for test db Category: Final Exam (300) Fully Scored.													
☐ Name Heckendorn, John Kronick, Scott Kowalski, Nicole Hambrecht, Betsy Takatsuka, Jim Rine, Jasper													

The bottom line in the grade crunching aspects of the software is that the program is flexible enough to do whatever you want to do in the way of calculating grades. Moreover, it is easy to get what you want. The program can do weighted averages, plain averages, delete the lowest n scores, and accommodate such variables as incompletes, dropped students, and excused absences. There are more options available for handling grades than one would probably use in any one class, but at the same time the program is not weighed down with marginally useful frills that would make it more difficult to learn. With regard to the second test, I spent several hours one evening trying to foul up the entry of data. The program is remarkably resistant to stupidity. For example, I could not enter more points on an exam than the exam was worth without having the program warn me. Nor could I impose an illogical cutoff for grade standards (i.e., it is impossible to establish a B as greater than or equal to 80% and then set a C as greater than or equal to 85%). The only errors that I could get away with were to put the same name in twice, and once I managed to erase the entire class roster. The program has a built in security feature that places an asterisk next to any score that has been changed after it was first entered.

My one serious complaint has to do with how the curve for a course is established. In this program, one chooses the score cut-offs for each letter grade (pluses and minuses are accommodated) and then the program calculates the final grades based upon the cumulative score or upon individual grades. If the curve does not look good, one can simply change the cut-offs and the program recalculates the grades and the curve. For many applications these approaches will be sufficient. However, in our large class, we always plot a curve of score versus rank with each student represented by one dot on that curve. In establishing the cut-off between a C- and a D+, for example, *Micrograde* is blind to whether the division separates students whose total scores differ by only a single point and heartlessly makes the cut. In contrast, Berkeley professors exhibit their infinite wisdom and compassion in such cases and look for

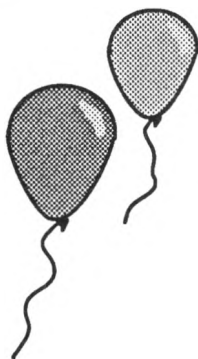
BMUG FALL 1986

the next logical breaks in the curve. If the class has been alert and laughed at all of the professor's jokes, he or she will probably choose the break on the downhill side to divide letter grades. Hopefully, the next release of *Micrograde* will incorporate some humanity and make this option available.



The last two features are two of the strongest points of *Micrograde*. The instruction manual is well-illustrated and lucid, although a bit redundant. If DOS software documentation had been written this well, I may never have tried a Mac. The most significant strength of this program, in my opinion, is the variety of ways in which printed reports of the data can be prepared. The program will print histograms of individual assignments or test scores and list the class either by name or student number with the corresponding grades. Alternatively, for courses such as ours in which a grade isn't calculated until after the final exam, the reports can list the students and their relative rank in the class on an individual exam, or in the entire class. The ability to obtain this type of data easily answers the final question of whether the effort was worth it. Given that our universities have complex rules of when a student can drop a course and not have the drop appear on a transcript, or options that sometimes let them repeat a class to change certain letter grades but not others, it is only fair that they be given as complete information as possible regarding their relative progress. *Micrograde* is a tool that should help in this goal, and I intend to try the program under real battle conditions this year.

Jasper Rine is on the faculty of the Biochemistry Department at the University of California, Berkeley. Besides his Macintosh, he enjoys yeast genetics and fly fishing.



balloons/Wood



All the Home Accounting Programs

Scott Beamer

What will a home accounting program do for you? Maybe it will help you clean up your act, and when it comes to money, almost all of us need to clean up our acts. It will help you balance your checkbook and keep it up to date. It will help you balance your accounts from your monthly statements. It will help you make, and stay, on a budget. It will remind you which bills you meant to pay when. It will even write checks for you (and address the envelopes if you use windowed envelopes). Best of all, it will be a big help at tax time. It will probably save you more than the cost of the program the first year. By the way, "Home Accounting" is a misnomer. These are actually "home bookkeeping" programs. If you want an accounting program, there are literally dozens to choose from, but they assume accounting skills to use them. If you can use a Mac, you can use any of these programs.

MacMoney (Survivor Software) \$79.95. Version 2.02. It is the current home accounting program of choice. It has the most elaborate features of any home accounting program on the market. Data entry is fast and intuitive, with separate forms for checks, credit cards, bank charges, and even adjustments. There is a memo line on all these forms, so you can leave a good paper trail to remind you later what you are doing now and why.

Report generation is extremely elaborate, allowing the user a maximum of customizing capabilities. There are so many choices, it's intimidating to use, but you will not need to generate reports until you have gotten used to the program for a while. By then, it will be easier. There are added features, such as mortgage analysis, IRA planning, and retirement planning. I find I hardly use these features, though I don't feel I paid much extra for them. The mortgage analysis is impressive, though and generates a complete thirty year amortization table.

MacMoney just got handier. **MacIntax**, the personal income tax program, has just announced that it will be able to import text files this year from many programs, including **MacMoney**, and allow you to mouse click the numbers into your tax forms. Next year, **MacIntax** expects to have this process automated so that **MacMoney's** annual report will automatically fill out your forms. Sound too good to be true? Yeah, but I got to try it. This program is fine for students, yuppies, hackers, families, in fact, most anyone. If you have a small business, charity, or user group, this may be all the accounting software you need.

The screenshot shows the MacMoney interface. At the top is a menu bar: File Edit Transactions Lists Plan Reports Graphs. Below it, on the left, is a 'Names' list with entries like McCab's, Montclair Children's Shop, Monthly Interest, Neiman-Marcus, Nordstrom, Northwestern Mutual Life, Oakland Scavenger, and P.D.A.E. In the center is a 'Scott Check' form with fields for Date (06/28/86), Check # (806), Pay To (Neiman-Marcus), Memo (Can't Afford it!), and Category (Merchandise). There are 'Save', 'Quit', and 'Split' buttons at the bottom of the form. On the right is a table of 'ID Codes' with columns 'ID' and 'Name', showing 'VE Versateller' and 'VO Volvo'. At the bottom left is a 'Categories' list with columns 'Category' and 'Type', listing items like Charity, Check Charges, Check Interest, Clothes, Computer Expense, EBHLD, Entertainment, and Estimated Tax.

MacMoney

The screenshot shows the MacIntax 'BEGIN NEW ACCOUNT FILE' dialog box. It has a menu bar: File Edit Select Maintenance Report Graph Controls. The dialog has fields for Title (Joint Accounts), Year (1986), and First Month (1). There are 'OK' and 'Cancel' buttons. Below these are two sections: 'Initial Set of Accounts' with radio buttons for Tax Preparation Accounts, Household Accounts (selected), Business Accounts, and No Initial Accounts; and 'Accounts Selected' with a list box containing Mortgage, Check Interest, Dividends, Gross Income, Interest Income, Auto Expenses, and Check Charges.

Dollars and Cents



Dollars and Cents (Monogram Industries) \$150/\$93. It is now out in version 1.4. This version is HFS compatible, has a mini finder and no copy protection. Upgrades are available at additional cost to registered owners. It opens and closes faster, but has no other apparent differences from earlier versions. This program has been among the most popular home accounting programs. It has sold over 100,000 copies in versions for various machines. I used it for a year, but what is needed is version 2. This is rumored to be available sometime next year.

In the meantime, Monogram has been busy with other projects. They have introduced a companion program for *Dollars and Cents* called *Forecast* that offers increased graphic and forecasting abilities. They currently have in place some hookups for home banking, but none in California yet. They hope to soon offer a tie in with a tax package so that your end of the year report will be automatically transferred to your tax sheets.

In terms of performance, I'd wait and see what version 2 is like. My biggest gripes about version 1.4 are the same as with version 1.3, that there is no memo line on checks and its report generating abilities are limited. I consider the memo line mandatory if one is going to leave an adequate trail of notes to remind one why one did things when looking back from a year or years later. Hmmm? Why did I write a check to Miss Toots LaRue for \$200? Look at the memo line and see "Party Favors". Oh, that's it!

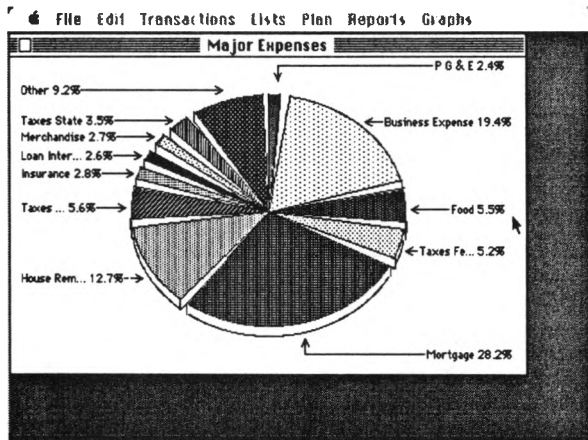
The reports are limited and offer inadequate editing capabilities. The part that irks me the most is that the budget features can't be turned off. I don't happen to use the budget features, so I find this very irksome, causing lots of blanks and spurious data in the reports and charts. For the impatient, you can buy version 1.4 now and upgrade later. To give this program a personality, I would say this one most feels as if it were written by an accountant. It's certainly reliable, but unimaginative... the manual, too.

The Home Accountant and Financial Planner for the Macintosh (hereafter to be referred to as *HAMac*) (Haba Systems) \$115/\$64. This is a product dead in the water. The programming team that wrote it no longer has a working relationship with Haba. There are no further upgrades for the product planned and no bug fixes. Though the program has some bugs, it actually works quite well. I prefer it to *Dollars and Cents*. The only reason I don't use it is that there is a better product available. The programming team that wrote *HAMac* have formed their own company and are offering what is essentially Version 2 (2.02 at last count) of the program. It is now called *MacMoney*.

The improvements of *MacMoney* over *HAMac* include: it will import *HAMac* data (but *HAMac* will not import *MacMoney* data), it invisibly looks for some data base bugs and will attempt to fix any found every time you open a file, and it will print out reports in ASCII text. This allows you to move reports to a word processing article, or to *Excel* for further processing, or to *MacIntax*.

ComputerWare includes a book on home financial planning (not Mac specific) for the \$64 price.

BMUG Fall 1986



*Home Accountant and Personal Financial
Planner*

Date	Num	TDate	Account	Type	Amount	Balance	Destination	Descri
01/08			AMEX	PAYMENT	325.00	325.00	PAYMENT	
01/20			AMEX	PAYMENT	55.00	55.00	PAYMENT	
01/02			AUTOLORN	PAYMENT	210.00	210.00	PAYMENT	
01/20			AUTOLORN	PAYMENT	210.00	210.00	PAYMENT	
01/15		01/15	CHECKING	EXPENSE	6.00	6.00	BANK CHG	
01/20			DINER	PAYMENT	30.00	30.00	PAYMENT	
01/11			MC	PAYMENT	100.00	100.00	PAYMENT	
01/20			MC	PAYMENT	25.00	25.00	PAYMENT	
01/02			MORTGAGE	PAYMENT	590.00	590.00	PAYMENT	
01/30			SAVINGS	INTERND	6.00	6.00	INTERND	
01/03	104	01/10	CHECKING	CHECK	33.90	33.90	FOOD	Albert
01/01			MORTGAGE	OPENBAL	80,000.00	80,000.00	OPENBAL	Americ
01/02			MC	EXPENSE	131.00	131.00	AUTO	Earl I
01/01			AUTOLORN	OPENBAL	4,000.00	4,000.00	OPENBAL	First
01/16			DINER	EXPENSE	30.00	30.00	FOOD	Maxwell
01/01		01/01	CHECKING	DEPOSIT	332.91	332.91	DEPOSIT	Open B
01/01			MC	OPENBAL	250.00	250.00	OPENBAL	Open B

OverVUE Personal Finance Template

OverVUE Personal Finance Template (ProVUE Development Corp.) \$39.95/\$34.95. It is very inexpensive, and if you already have and like *OverVUE*, this may be an adequate solution for you. The template has almost all the features, but is not as agile, as a pure home accounting program. This is somewhat offset by the fact that *OverVUE* will allow the advanced user to customize these templates to their heart's content. For that person, this program provides great power, maximum customizing abilities, and ease of use. A word of warning... *OverVUE* recommends that you do not try to customize the structure of the template as it will interfere with the Macros and hence, the operation of the program. If you attempt it, do it only on an expendable copy.

This program also lacks the Memo line on checks. The reports are clumsy to implement and slow. There are charts available. Data can be exported to other programs as a text file, for instance, *MacIntax*.

Electric Checkbook (State of the Art) \$79.95/\$39 This is the junior member of the group. It does what it does well, but it lacks the features of the other programs. This might be the program of choice for someone who is more concerned with how much of which bills to pay this month than say, retirement income planning.

The basics of the program are well implemented. Error correction is especially good, but the manual is only 44 pages...with a fourteen page addendum...with a one page addendum. The fourteen page addendum covers such items left out of the manual as credit cards and mortgages. The one page addendum tells: of a new feature allowing checks to print the payees address (a feature the other programs have except *OverVUE*), the upgrading of the finder to 4.1, that **version 2.0 will not accept data from the previous versions**, (all version 1.0 data must be rekeyed (Argh!!!)), and finally, all tech support is by mail only. To me, it appears that State of the Art (that's their name, honest... and based on these features, a sorry state it is!) has spent more time worrying about **copy protection** and keeping you from xeroxing the manual (it's printed in grey and blue) than paying attention to the value of their product, especially compared with the competition.

BMUG FALL 1986

File Edit Task Categories Bank Accounts Help

Reconcile Account: LAST NATIONAL BANK

Chk	Date	Amount
W th	02/19/85	1,000.00
Dep	02/26/85	1,100.00
Adj	03/05/85	15.00
Dep	03/15/85	900.00
BrkCg	03/16/85	10.00
Adj	03/22/85	132.32
BrkCg	03/22/85	6.00
W th	03/27/85	75.00
Int	03/31/85	23.94
Dep	04/01/85	1,250.00
W th	04/22/85	32.00
W th	05/01/85	50.00
00100	03/25/85	79.45
00101	03/01/85	225.00
00102	03/15/85	0010
00103	03/28/85	20.00
00104	03/31/85	25.00
00105	04/12/85	12.00
00106	04/12/85	23.83
00107	04/15/85	2,118.00
00108	04/14/85	464.32

Statement Balance 5,025.10

Check Number

Outstanding...

Checks (-) 2,675.99

Deposits (+) 1,250.00

Adjustments (-) 82.00

Net Balance 3,517.11

Current Balance 3,684.11

Out of Balance By -167.00

Accept Report Purge

Options Edit Transaction Account Budget Graph Report Help

System Information

Enter your name and address as it should appear on your reports:

Name: J. K. Lasser.

Address: 2160 Stoned Road

Berkeley, CA 94705

Report Page Length: 60 lines.

Number of Accounts: 6 Number of Budgets: 7

Number of Payments: 1 Number of Deposits: 1

Number of Charges: 1 Other Transactions: 1

Available Memory: 212,274 Available Disk Space: 7,971,840

Electric Checkbook

Your Money Manager

J. K. Lasser's YOUR MONEY MANAGER (Simon & Schuster) \$99.95. This program was released this week. It was received literally too late to review. I have had about two hours to fiddle with it, and based on that time, and reading the manual, here are my impressions.

The program has two big selling features. One is that J. K. Lasser has produced the most popular personal income tax guide for almost fifty years. There is instant recognition and a great deal of respect associated with this name. The other feature is that the program is designed to work directly with a companion program, **J. K. Lasser's YOUR INCOME TAX**.

My first impressions indicate that both these features have worked against the program to some degree. In the first case, while Lasser's tax guide is printed on cheap paper in a large densely worded workbook with exhaustive indexing, **YOUR MONEY MANAGER** comes in a hardback manual with a garish oversized cover. Inside is a laughably small manual, luxuriously printed, that never gets past the first time tutorial stage of dealing with the program. There is no index! Fifteen percent of the manual are four pages of full sized screen dumps showing each of the pull down menus in the main menu bar. Any but the most timid of Mac users will be impatient if not insulted with the spoon feeding style of this manual.

The second feature, the connection to **YOUR INCOME TAX**, causes another set of problems. The idea is to allow you to shift all your bookkeeping data from **YOUR MONEY MANAGER** to **YOUR INCOME TAX** with the click of a button, thus filling out your income tax forms almost instantaneously. I have not entered enough data to try this out, but already I have found the principal drawback to such a scheme. Your personal files must be arranged according to the precise list of categories needed to effect the transfer. Other than listing the categories along with an I. D. code, the program does not help with this.

Worse yet, the program requires you to enter a four digit code for every transaction. There are dozens of these categories to memorize. You can have a screen of the codes visible, but there is no implementation of a double click data entry system that some of the other programs have.

Don't get me wrong. The package looks good and I'm certain it works. It is written in MacForth. It has nice windows and controls, and is impressive in its speed. It is fairly full featured, and has lots of command keys. My favorite is Command-E that closes all files and does a complete shut down. The rest of the programs should take note of this one!



It will print checks (free samples enclosed), make automatic payments, print charts, and do several reports, including a personal financial statement. There is tech support by phone, and it is not copy protected.

My first impression tells me that this has the basis of a nice accounting-bookkeeping package, but version 2 is needed right now.

For a fuller evaluation of these products, see October Macazine.

Products:

MacMoney
Survivor Software
11222 La Cienega Blvd.
Inglewood, CA 90304

Dollars and Cents
Monogram Industries
8295 S. La Cienega Blvd.
Inglewood, CA 90301

Home Accountant and
Personal Financial Planner
Haba/Arrays
11223 South Hindry Ave.
Los Angeles, Ca 90045

Personal Finance Template
OverVUE
ProVUE
222 22nd St.
Huntington Beach, CA 92648

Electric Checkbook
State of the Art
3191-C Airport Loop
Costa Mesa, CA 92626

YOUR MONEY MANAGER
Simon & Schuster
1230 Avenue of the Americas
New York, New York 10020

Objective Data

	MacMoney	HAMac	Dollars & Cents	OverVUE	Electric Checkbook
List Price	\$74.95	\$99.95	\$150.00	\$39.95	\$79.95
Discounted	?	\$62.00	\$81.00	\$34.95	\$42.00
Minimum System	512k	128k + ext	128k + ext	128k + ext	128k
Num of accounts	200?	200?	12	limited	10
Num of Categories	200	200	120	by RAM	100
Num of transactions	3000 to 6000	3000	2000	3500 ca	500
Size of Program	255k	208k	228k	129k*	60k
Copy Protected	no	no	yes	no	yes
Help Menus	yes	yes	yes	no	yes

* will work only with OverVUE 2.0 which is another 145k

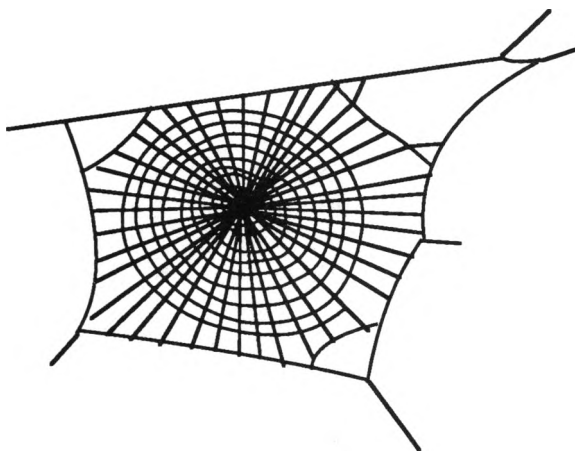


BMUG FALL 1986

Subjective Data

(highest rating = 1)

	MacMoney	HAMac	Dollars & Cents	OverVUE	Electric Checkbook
Easy to learn and use	3	3	4	2	1
Follows Mac Interface	2	2	3	1	4
Flexible	1	2	4	3	5
Full Features	1	2	3	4	5
Report Quality	1	2	4	3	5
Charts Quality	1	1	3	2	none
Speed	5	4	1	2	3
Technical Support	1	2	4	3	5
Manual	1	2	3	5	4



Wood/SpiderWeb



Mini Reviews:

Why Don't They... Hey, Wait -They just did!

Scott Beamer

MacInUse. That's exactly how it happened. I was talking to Softview on the phone, and they mentioned their soon to be released new program. Exactly the program I had just described in a "Why don't they..." for this newsletter. Picking myself up off the floor, I asked for a copy; and now I am back at my Mac, ready to tell you about it.

Remember Softview? Name ring a bell, but you can't quite place it? You'll know the name again in a few months. The Softview people produce MacInTax. Their new program isn't for taxes. Though you could use it for tax records. It's called MacInUse. It will list for \$49. They're hoping to release it at the Dallas Expo. It's a desk accessory. It installs itself in your system through an INIT. Don't worry about that. You just click the icon; it does the rest. MacInUse creates a time log for the period your Mac is on, recording how long you use each application.

I can't say how many people have the same compulsion about time that I do. To me, this program seems like a sure winner. With it, you can produce records to justify how you depreciate your computer expenses, or to do client billing. You can see what your kid was up to while you were out last night. This is a general purpose tool; figure out the use that suits you. It will certainly be tax deductible.

MacInUse shows the intelligence and careful crafting I have come to associate with MacInTax. It can be turned on and off without removing the application. You can save a file, and start a new one, resetting all the clocks. In visible mode a dialog box allows annotation of the file, with buttons for Business or Personal, and a space for text. Alternatively, the police functions can be enhanced by making the program invisible, so no one else will know it's there. It will print out its files. It comes with templates to open the files in MacWrite or M. S. Word. It has a macro for Excel that will graph your time use from the MacInUse file.

A few day's use has revealed some shortcomings. It is slick, but I would like MacInUse to log the time under individual file and document names rather than just the application name. I also think the dialog box should look like the file does and should include a display of the time just logged. On my hard disk, it even logs my boot up disk. This is awkward as it requires clicking away the dialog box that would normally be used to annotate the MacInUse file. Still not perfect, but what a great program idea!

MacInTax (Softview)

It's getting towards that time again. Tax time. April 15th. MacInTax remains the only tax program worth considering, unless your needs are very modest, or you like to tinker with Excel templates. Susan Morgan, co-owner of Softview, promises the final release for 1986 MacInTax for the San Francisco Mac Expo in early January! They guarantee not to be late this year. A prerelease update will be available, but only to registered owners, in November.)

Some changes are in store. The main disk still lists for \$99.95 but will now include all the federal and business forms. The Business Supplement is eliminated. Updates will be \$45, on special this year for \$30; California forms are an additional \$45, (\$15 update).

There will be converter routines on all disks. The federal converter offers two options. One will convert the permanent information, such as name and Social Security number, from the 1985



BMUG FALL 1986

forms to the 1986. It also transfers and updates Schedule G (Income Averaging). The other option will convert all information on the 1985 forms so that the 1986 MacInTax can read the files.

The big news for this year is that MacInTax will accept text files from Excel, Jazz, many full accounting programs, whatever. It works especially well with MacMoney; it does not work, this year, with Dollars and Cents. You will see your transferred files on screen in a window. Select some data, then point on the tax form to where you want it and click to complete the transfer. In 1987 MacInTax will do this transfer automatically with selected programs including MacMoney, Dollars and Cents, and several other large accounting programs. A professional accounting program from MacInTax is also in the works.

MacZap 4.3 & Lifeboat.

Micro Analyst has just done an extensive workover of their MacZap package. Their ideas seem good, but the results are disappointing. Everyone I know reaches for Copy II Mac when they need a copy, even when MacZap Copy is available. I don't expect to see this change, as MacZap hasn't changed much in essence. There is something very unMac-like about the screen and controls.

The utilities now handle HFS and 800K disks. Fast Disk Copy is a new addition to the disk that facilitates fast disk copying, especially if more than one copy is being made. It requires some disk swapping.

Micro Analyst has a version 2 of their book on software protection (\$33). This book is not specific to the Mac. There is a new book on the details of the difference between MFS and HFS (\$18). I haven't seen either of these books, but if the MacZap manual is any example of their writing style, look out! The manual is one of the most opaque I have ever tried to read.

Lifeboat is a BBS run by Micro Analysts for registered users of MacZap ((512) 926-9582). It is apparently intended to allow registered users rapid access to new patch files without having to wait around for the next update of MacZap. The BBS includes section for: File, Floppy Disk, and Hard Disk recovery, Patch File Downloads, Developer's Information and Support, Public Domain Programs, and Messages.

A good idea: but results remain to be seen. The BBS started up in August, but no one is there. I have called in several times over six weeks checking for new files or messages. In all that time, there was only one new file or message left. My messages to sysop were never answered. The patch file list remains exactly the same as the one with the latest MacZap upgrade disk. (Save for some minor patches to fix bugs in the various MacZap utilities.)

Why Don't They?

Does anyone know of a database mailing list of the United States Congress? If there isn't one, why doesn't BMUG make it a public service project to make one? It's only about six hundred names. We could split up alphabetically for typing, and later join the lists, if we do it in OverVUE. The information ought to be in the public library. It might become BMUG's most popular disk, a good attention getter at shows. It would be nice on a BBS, as well. I can think of some very good uses for this. Can you? Ought to work great with M.S. Word mail merge. (See Piracy and Rentals).



spelling checker, included in it as a standard feature, is very good, except that the dictionary is too small. It is the second smallest among all programs considered here, and it recognized the second fewest of the 500 Hard Words list. However, it placed first in guessing the correct spelling on the 100 Wrong Words list! It is also the fastest guesser, and its one click corrections tied for fastest with *Spelling Champion*.

The spelling checker's skill at guessing is due to multiple guesses, but more to its algorithm. It has four criteria: transposition, deletion, change, and insertion of up to two letters in each combination. Certainly none of the other checkers have as elaborate a routine for guessing, yet *WriteNow* is also the fastest of the true guessers, (the others being those that only turn to the closest alphabetical fit). In normal use, this is a very fast, slick way to spell check, but I still want the larger dictionary.

The manual is arguably the best I have ever seen for a software program, written by Caroline Rose (of *Inside Macintosh* fame) and Lisa Kelly. This manual is for *WriteNow*, the word processor. There are only four pages on the spelling checker, but they should be a model for forthrightness and clarity to all the other manuals.

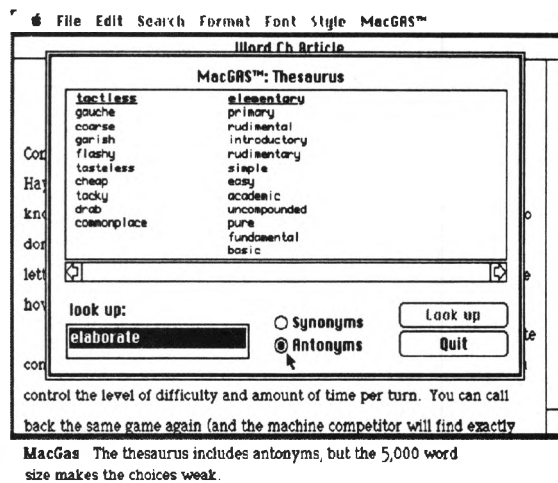
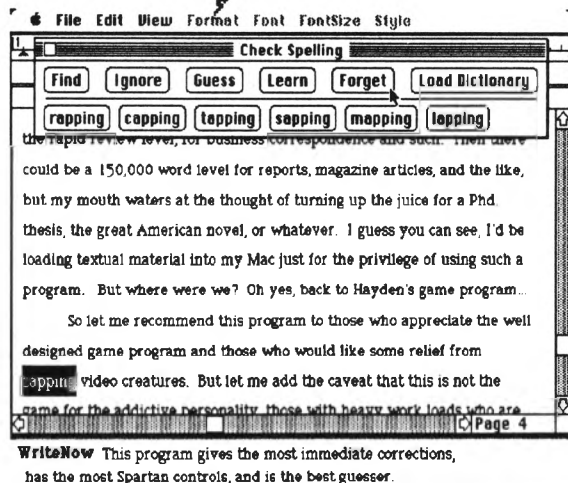
I cannot recommend that anyone buy this program as a spelling checker. It is primarily a word processor. I include it in the article because it is a very good spelling checker. Those who are shopping for a new generation word processor and a spelling checker may be able to make one purchase instead of two.

The spelling checker controls are very Spartan, yet do the job just fine. This part of the program is as fast and unobtrusive as you might hope a spelling checker could be. When you are word processing in *WriteNow*, just place the cursor, and one command key (or a mouse click) starts spelling checking. The process is extremely terse and intuitive.

The program does not check capitalization, punctuation, or for double words (e.g. and and), but I seldom use these features. There is no window showing entries in the dictionary. You may add and delete words from the dictionary, and you may have multiple versions of the dictionary for different uses. At a later date, I hope they might offer an enlarged dictionary as an optional accessory. That condition met, this might be all the spelling checker I require.

Words it challenged in the bench tests but shouldn't have : Dr., Mr., touchier, nomad. It suggested I respell "algorithms" with "aogorythms" (sic).

BMUG FALL 1986



MacGas

MacGas vers. 1.06 (EnterSet & Applications Unlimited) \$99.95/\$55. 80,000 word dictionary. Two disk program. No on screen help. No tech support mentioned in the manual. An address is listed on the title page, but no phone number. It works as a DA, and supports *MacWrite*, *MS Word*, text documents, *Jazz*, and many other applications.

The manual is short and leaves out a lot. The one page addendum for the newer version (1.06) is essential, but still not enough.

MacGas stands for "Mac Glossary and Spellchecker." There is an interesting collection of icons on the disk. These include two dictionary icons, a glossary, a thesaurus and a gas pump. The thesaurus is for 5,000 words. This is a bit anemic, but it's included in the price of the program while *MacLightning's* 45,000 word thesaurus will put you back \$50. The glossary is a useful accessory that allows you to define abbreviations that the word processor will write out in full. They can include the date and time. *MS Word* also includes this feature.

Another feature is Word LookUp. This feature is available any time in any program (including the finder) in which you have access to the Apple menu. It will look up any word you type into the dialogue box presented. It is to confirm the spelling of a word you're not sure about. You can type multiple guesses, one after the other, or scroll from your first guess, but if you don't type the word correctly (or if the word isn't in the dictionary), it's not going to help you at all. The program does not guess the spelling of challenged words well. It only turns the word list alphabetically to the closest fit, the same as *Spellswell*.

The two dictionaries are a basic small one, and an enlarged one. Personally, I wouldn't use the program with the small one. I tried it, and it was very weak. This is the only program whose performance on the tests didn't meet expectations from the claimed dictionary size. It got the same score on the Hard Word list as *Spellswell* version 1.2c which claimed a 60,000 word dictionary.

In general, the program works well and has an interesting group of features. It does one click correcting when you find the word in the dictionary, but because it is a poor guesser, it is usually



fastest to retype the word in the **Checking Word** box. Another significant problem is that though *MacGas* encourages you to add words to a user dictionary, it provides no means of deleting words in case you make a mistake. All these programs should allow you to delete words from the main dictionary as well, in case they have made a mistake.

The manual addendum warns to **Cancel** the spell checker before trying to edit the document directly. If you don't, it will bomb. This caused the only bomb during all of this testing. That's fairly remarkable considering I had four spelling checkers as Desk Accessories in the same system file and seven spelling checkers, with all their assorted dictionaries, on the same volume of my hard disk.

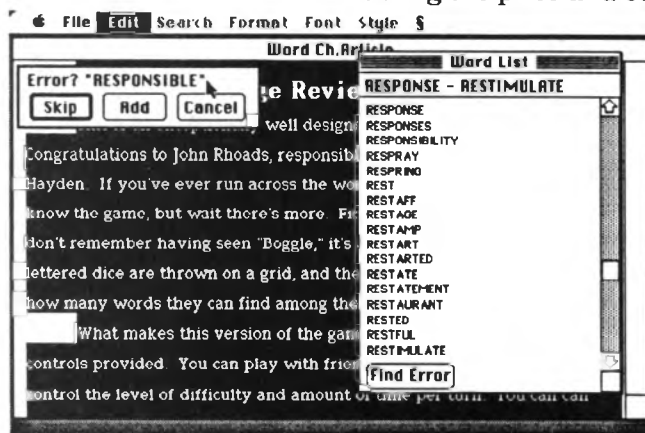
Words it challenged in the bench tests but shouldn't have: fewer, iodine. (Dictionary included "pa'ange." I haven't figured out what that means yet).

Liberty Spell II

Liberty Spell II (DataPak Software, Inc.) \$59.95/\$42.00. 30,000 word dictionary(101K), program 17K. No on screen help. Tech support by phone! Works with *MacWrite*, *MS Word*, text files, and many other programs. Specifically designed to work with *Executive Office* and *My Office Plus*.

Liberty Spell II compares poorly with the other spelling checkers reviewed here. I believe this is because the program was originally intended as an accessory for the programs *Executive Office* and *My Office Plus*. I have not tried it in that environment, but the manual mentions this obliquely, and I can't imagine a reason for this program's existence if it isn't so.

This program operates as a desk accessory, so it is readily available in many different programs, but to activate the spell checking function, you must cut and paste (using the word processor's controls) the document into the **Word List**. To correct misspellings, you must either do it manually within the word processor or make a macro to use the search and find controls of the word processor. After every correction, to reactivate the spelling checker, you must repaste the document into the **Word List**. Considering the price and compared to the



Liberty Spell II What can we say? This is the weakest product reviewed.

competition, this primitive system of controls is just not acceptable. Apparently these primitive controls are replaced by regular menu items in *Executive Office* and *My Office Plus*.



The dictionary is by far the smallest among these programs. There seems no excuse for it. On the disk with the program is a separate folder intended as a gift for a purchaser of this program. It contains a public domain game (*Billiards Parlour*) taking up 92K on the disk. Why didn't they use this space to enlarge the dictionary and improve the quality of the program? Were they playing *Billiard Parlour* instead of programming?

And here's another item in the "you already guessed it" category. *It's copy protected*. It's the only one reviewed here that is. This is the smallest, poorest quality program among those reviewed here, yet it is not the least expensive. It is the only one I would say "Don't buy it unless you are absolutely certain you know what you're doing..."

A final note here... Other reviews are not nearly as harsh on this product. My opinion seems in a distinct minority. *MacUser* gives it four mice, and *MacBriefs* mentions it in two relatively favorable article excerpts. Even *MacJournal* wasn't that harsh on it. Why the difference of opinion between this article and those? Personally, I think they're wrong. The poor quality of most magazine reviews is much of what prompted me to write this article.

Words it challenged in the bench tests but shouldn't have: timer, mouth, questioned, waters, speller, creature, massive, etc. etc.

Thunder

Thunder, (Batteries Included). \$49.95. 50,000 words dictionary.

Here's some info for those of you in BMUG who have heard rumors about this program. Some may have seen the full page color ads as well. They claim to be three products in one. Don't be misled. I have not seen or used this product, but I believe it has no features radically different from others in this review. I talked to the programmer for about fifteen minutes, recently. They are now hoping for a late October release. The name is, as the product, rather Canadian (as in Thunder Bay), but is more clearly aimed at whom they consider their main competition in the market. The big drawback is that the dictionary will only be 50,000 words. When I tried to point out the trend was definitely towards larger dictionaries, he countered that the quality of the selection is more important than the count. His was done by another Canadian company and included every word in a Canadian paper for the last ten weeks, something like that.

The good news is that the dictionary includes both American and Canadian (English) alternate spellings. This may be important for some, especially Canadians, as most of these spelling checkers are unrepentantly American. Three of seven recognized "salsa," but only one (*Spelling Champion*), recognized "cheque."

Thunder opens as a DA, and will spell check interactively or for an entire selection. It makes its corrections through the clipboard. The programmer claims that this permits his program to actually correct the spelling in more programs than any other. The drawback to this is that only in *MacWrite* will the correction retain any formatting, including font. For the rest of the programs, including *MS Word*, the correction will appear in plain text, even if the document is not.

It has a word count (only two programs in this review don't) and a glossary. *MacLightning* and *MacGas* also have glossaries. It seems that a glossary is more useful in the word processor. M. S. Word has a glossary.



Thunder seemed very proud that it worked in the interactive mode. In this review, *MacLightning* and *Liberty II* also work in the interactive mode. *WriteNow*, *MacGas*, and *MacSpell+* are available from within the word processor, and can check even a single word.

With interactive checking, your Mac beeps every time the spelling checker doesn't recognize the combination of letters you are currently typing (well, almost currently. You don't have to be a very fast typist to get way ahead of it). You may continue typing to finish the word, sentence, or idea, but if the machine doesn't recognize a second letter combination before you taken action on the first one, the first one will be forgotten. Personally, I never use this style checking. I found I had little patience with the incessant beeping even to review it for this article. In talking with others, I found no one who uses this mode. The constantly ringing bell and the distraction of needing to take action on the spelling causes one to constantly lose one's place or train of thought. With a 50,000 word dictionary, as in *Thunder*, the bell can be expected to ring about five times for every time a spelling correction is actually required. So who uses the interactive mode? Perhaps people who type very slowly and infrequently are the only ones.

The programmer at Batteries Included was very friendly and cooperative, saying he knew some people from BMUG. Unfortunately, the business person I talked to did not. He explained that their policy was to never give promotional copies, even beta, to freelance press. Besides, he had never heard of BMUG. "Send us a copy of your newsletter," he said, "and we will consider sending you a copy of the program." Sometimes, that Canadian border seems a little thicker than I think it is. This was the only one among the eight companies I contacted for this article that did not send a program and promotional literature.

Judging only from my telephone interview with the company mentioned above and their ads, it would appear that this spelling checker is entering the market with too little, too late. We can count on a low price, lifetime \$10 upgrade policy, and no copy protection, as with other Batteries Included products.

Spelling Champion
Champion Swiftware
5201 South Hill Drive
Madison, Wisconsin 53705

Spellswell
Greene, Johnson, Inc.
15 Via Chualar
Monterey, California 93940

WriteNow
T/Maker Co.
1973 Landings Drive
Mountain View, CA 94043

Liberty II

MacLightning 2.0
Target Software Inc.
14206 S.W. 136th Street
Miami, Florida 33186

MacSpell+
Creighton Development, Inc.
16 Hughes Street
Irvine, California 92718

MacGas
EnterSet
410 Townsend St. Suite 408 B
San Francisco, California 94107

Thunder

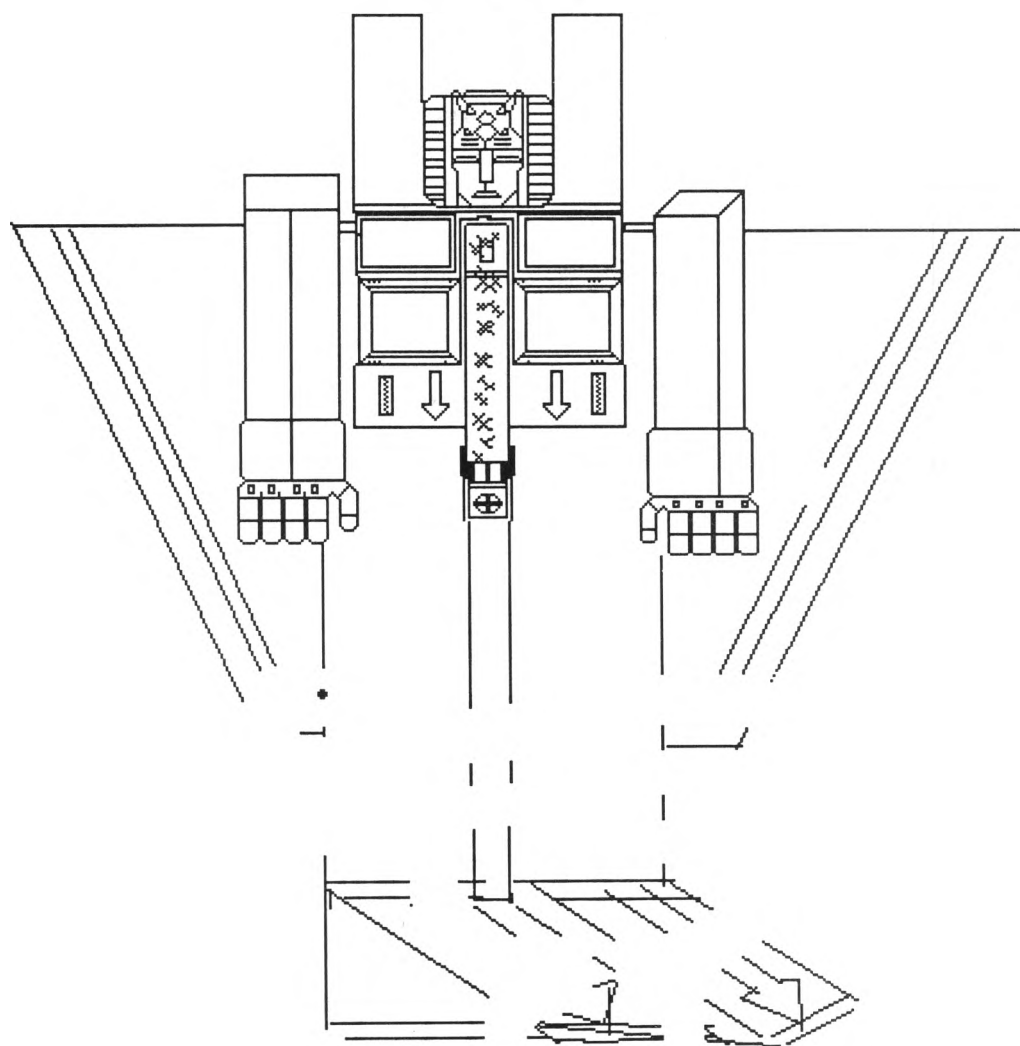


BMUG FALL 1986

DataPak, Inc.
14011 Ventura Blvd. #507
Sherman Oaks, CA 91423

Batteries Included
30 Mural Street
Richmond Hill, Ontario
Canada L4B 1B5

Starscream



Star Scream/ Zambrano



Fonts

Richard S. Palais

Fonts are the foundation of word processing (WP). The purpose of all WP software is, roughly speaking, to take symbols from various fonts and to arrange them appropriately into lines, paragraphs and pages. However the concept of a font is so complex and multifaceted, involving as it does questions of aesthetics, psychology, history, and technology, that I can only give a brief introduction here and must refer the reader who is interested in more detail to the excellent written version of Don Knuth's 1979 Gibb's lecture *Mathematical Typography* (Bull. Amer. Math. Soc. (New series) 1, March 1979) and the many references given there.

In this age of computer word processing and typesetting, a font is usually understood to mean a set of up to 256 related "symbols" or letter shapes. The complete description of a font consists of two related data structures: the font metrics, which gives the dimensions of the boxes into which the symbols fit (as well as some more esoteric things called ligatures, kerning, and italic corrections), and the actual geometric description of the shapes of the symbols that go in these boxes. We will consider these one at a time.

Font Metrics and Character boxes.

If you look closely at text typeset in some particular font you will notice that it consists of rows of characters that seem to rest on imaginary lines, called the baselines of the rows. For the most part the lower case letters extend above the baseline by a fixed height, called the x-height of the font in question. But there are some exceptions to these generalities: some lower case characters, namely "g", "j", "p", "q" and "y" have pieces called descenders that extend below the baseline, and others, namely "b", "d", "f", "h", "k", "l" and "t" have ascenders that extend above the x-height line. The height of the highest ascender is called the ascent or height of the font (or sometimes its cap-height since, in so-called Modern fonts at least, it is also the height of the capitals). We shall also take this to be the height of the top of a character box above the baseline. The depth below the baseline of the lowest ascender is called the descent of the font.

Note that the x-height, ascent, and descent are global parameters of a font. In so-called mono-spaced or "typewriter fonts", the width of a character box is also global to the font, while in proportionally spaced fonts it is local to each character. Needless to say, it makes it much easier to write a formatting program if one has only to deal with mono-spaced fonts, and almost all IBM-PC WYSIWYG formatters take advantage of this simplification. However, proportionally spaced fonts are more legible, and for this reason the ability to handle proportionally spaced fonts was built into the Macintosh ROMs.

The sum of the ascent and descent of the font is called its "body size"; the nominal point size of a font is a somewhat arbitrary dimension with no precise definition, but usually it closely approximates this body size. By the way, since we will be using the printers' units of points and picas, you should know that one inch is seventy-two points or six picas.

When a type formatting program makes a row of character boxes (i.e. a line of type) it aligns their baselines and (usually) it butts the left edge of one box up against the right edge of the preceding one. Note this means that the intra-word spacing is fixed by the font; in fact normally a character fits flush left in its box so that the space between it and any next letter in a word is the space between the rightmost point of the character and the right edge of its box.

Actually, characters do not necessarily fit inside their associated boxes! In particular characters in italic or slanted fonts will frequently lean outside the right hand edge of their boxes.

When a typesetting program arranges lines of type into a page the distance between a pair of adjacent baselines is called leading. Text is easier to read if the leading is very nearly uniform and, since it also looks better, an effort is usually made to insure uniform leading. The most primitive way to arrange this is to simply butt the top of one row of character boxes against the bottoms of the preceding row. This gives what I shall call the default leading of the font. Notice that if the bottom edges of character boxes in a font were taken to coincide with the descent line, then the default leading would equal the body size and descenders would touch the tops of capitals and ascenders just below them, giving a crowded look. In fact, traditionally leading is taken about twenty percent greater than body size; "ten on twelve" (i.e. ten point body size with twelve points of leading) or "twelve on fourteen" being two popular choices. For this reason the bottoms of character boxes (the depth) is usually taken one or two points below the descent line in text fonts. However mathematical symbol fonts are usually "set solid" (i.e. with leading equal body size) so that it will be possible, for example, to make large brackets by fitting together pieces on adjacent lines.

You may wonder what happens if you mix several fonts on the same line. The obvious answer is correct: the top or bottom of the line is given by the maximum top or bottom for any font in the line. Similarly superscripts in a line will move the top of the line up and subscripts will move the bottom down. This means that in mathematical typesetting, where font mixing, superscripts, and subscripts abound, default leading would lead to very uneven and ugly vertical spacing that is hard to read, and a good technical word processing program will use a vertical spacing algorithm other than default leading.

This may all seem somewhat abstract, so here are a few pictures which I hope will illustrate and clarify the above concepts. First a diagram of a character box (Fig. 1.); the point O is called the origin or reference point and plays a role in the more sophisticated systems like $T_E X$.

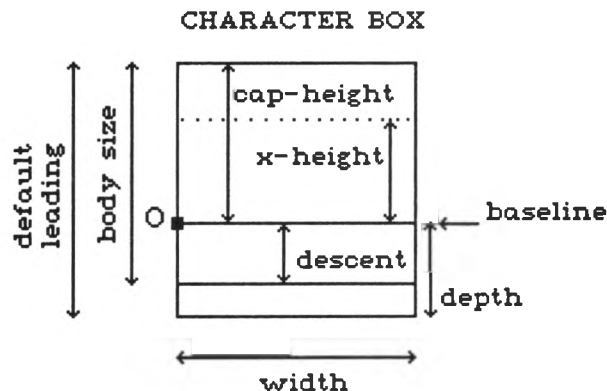


Fig. 1.



And here is a picture of a particular character in a particular font (the author's approximation of a lower case "g" in the Times Roman font). We show it alone and in its box (Fig. 2), and a slanted version of the same character (Fig. 3). Notice how it slopes a little outside its box!

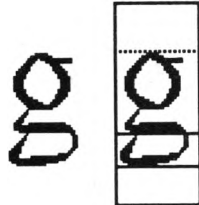


Fig. 2

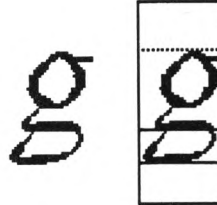


Fig. 3

Since all the characters in a slanted font are slanted at the same angle, intraword spacing is not affected. The difficulty comes in the spacing between words where there is a change from slanted to non-slanted text. In fonts designed for sophisticated typesetters the font metric file will contain so-called italic corrections that tell the system how much extra interword space to add to overcome this problem. In a WYSIWYG system the italic correction must be added by hand by the user.

The algorithm mentioned above for determining the spacing between pairs of adjacent letters in a word is clearly rather crude—it depends only on the first letter of a pair, and if you have had any calligraphic experience you know that for good looking text you must also take the shape of the following letter into account. The process of adding a positive or negative correction to the spacing between adjacent letters to make the spacing look more even is called kerning. For example the word "Type" looks better if the "y" is kerned left under the "T". Most WYSIWYG systems consider this correction not worth the effort, but again the top-of-the-line typesetters will put kerning information in their font metric files for every pair of characters that need them.

And what are ligatures? Look carefully at a fine book and you will notice that certain sequences of characters are replaced by a special alternate character. These are the pairs "fi", "ff", "fl", the triples "ffi", and "ffl", and sometimes the diphthongs "ae" and "oe". The quality typesetters like $T_E X$ will have such ligature information built into their font metric files and take care of them automatically, while the WYSIWYG systems almost universally ignore ligatures.

Kerning and ligatures may seem like precious fine points, and clearly whether they are worth the trouble depends both on the application and on personal taste. Here are a few examples in large type. First "Type" both with and without kerning (I added extra kerning to what $T_E X$ gave automatically to get what my eye told me was ideal) (Fig. 4.) and the words "fit", "affine", and "affluent", both with and without ligatures (Fig. 5.)

fit fit
Type affine affine
Type affluent affluent

Fig. 4.

Fig. 5

And how is all this theory related to the practical choices to be made in word processing? If you use a WYSIWYG word processor a decision you will often have to make is how wide to choose your column width and how much leading to use. The main criterion for this choice should of course be the ease and speed of reading the resulting text. A lot of experimentation has gone into this question and the general wisdom is that the width of two copies of the lower case alphabet of your principal text font is close to ideal for a single column of text. In a twelve point type this would give you a column width of between four and one-half inches for a very tight font to five and three-quarters inches for a loose one. I would recommend the choice of a proportionately spaced, twelve point, serified font with a "two alphabets width" of at least five and a quarter inches. Set your left margin at one and a half inches and your right margin at seven inches. As for inter-line spacing, for text with a lot of formulas interspersed I would recommend "double spacing", i.e. about twenty to twenty-four points of leading (with reasonable top and bottom margins this will give about twenty-seven lines per page). This may seem like too much white space, and I'll agree that one and one-half spacing (fifteen to eighteen points of leading) looks better (it's what I use for letters) but at that spacing a subscript j may overlap a superscript l below it. A final formatting choice is whether to have fully justified text or to use "ragged right" margins. This is mainly a matter of taste: ragged right is said to be slightly more readable, because of the evenness of the interword spacing, but many people prefer the neater appearance of full justification.

Font Shapes

So far we've said very little about the detailed letter shapes in a font. In fact the qualities that make for a beautiful font are remarkably subtle and have been painstakingly learned over hundreds of years. Also, with each change in the technology of printing, corresponding changes must be made in the design of old fonts and whole new families of fonts must be designed to maintain an optimal appearance for the text produced by the new methods. These are interesting questions but their general consideration would take us too far afield, so I will just say a little about the problems raised by the recent change from the older cold and hot metal type technologies to the current digital or raster technology of computer typesetting.

The first thing to realize is that the angular resolution of the human eye is approximately one minute of arc. Stated in terms of spatial frequency this means that at a reading distance of one foot the eye will resolve 250 lines per inch, but will see lines closer together than this as a gray blur. It then follows from the Nyquist theorem that to avoid "aliasing" or "jaggies" in a rasterized image (e.g. the staircase appearance of a diagonal line) the spatial sampling frequency must be on the order of 500 per inch, and this is indeed the lower limit of what is



considered acceptable photo-typesetting. In fact, quality typesetters have a resolution of at least 1000 lines per inch. This is because of what I call the kite string effect: the eye can see objects far smaller than it can resolve. A kite string at one-hundred yards subtends less than one-tenth of a minute of arc, yet is clearly visible, and similarly a good typesetter will set quite visible hairlines far thinner than the ability of the eye to resolve. Now for an eight and one-half by eleven inch page, 500 pixels per inch translates to a total of about twenty-four million pixels, or three megabytes to store one page. And if you want a video monitor which is flicker free (say 60 scans per second, non-interlaced), to show a full page at this resolution would require a processor with a clock speed of about 1400 megahertz, about two orders of magnitude faster than the chips used in current microcomputer technology. One common compromise is to show only a partial page on the screen. Another is to ignore reality and define "high" resolution for a video monitor to be about seventy-five pixels per inch, and for a dot matrix printer as about twice that. Obviously a great deal of aliasing is going to be inevitable at these resolutions and it takes very careful design to produce an even halfway decent screen font. Fortunately at the laser printer resolution of 300 pixels per inch it begins to be possible to build some semblance of quality into fonts, and while one misses the crispness of a fine font at high resolution, at least eye strain is less pronounced.

To give some idea of the degradation of quality at various levels of resolution I have placed side by side below my 12 point Times Roman lower case "g" as it might look if digitized at 2000 pixels per inch and at 300 pixels per inch (LaserWriter quality), both then magnified by a factor of twenty-five (Fig.6.), and how it might look at the same magnification when digitized as a screen font at 72 pixels per inch (Fig.7.).



Fig 6.

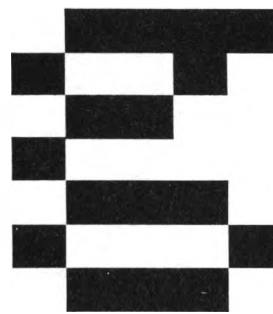


Fig 7.

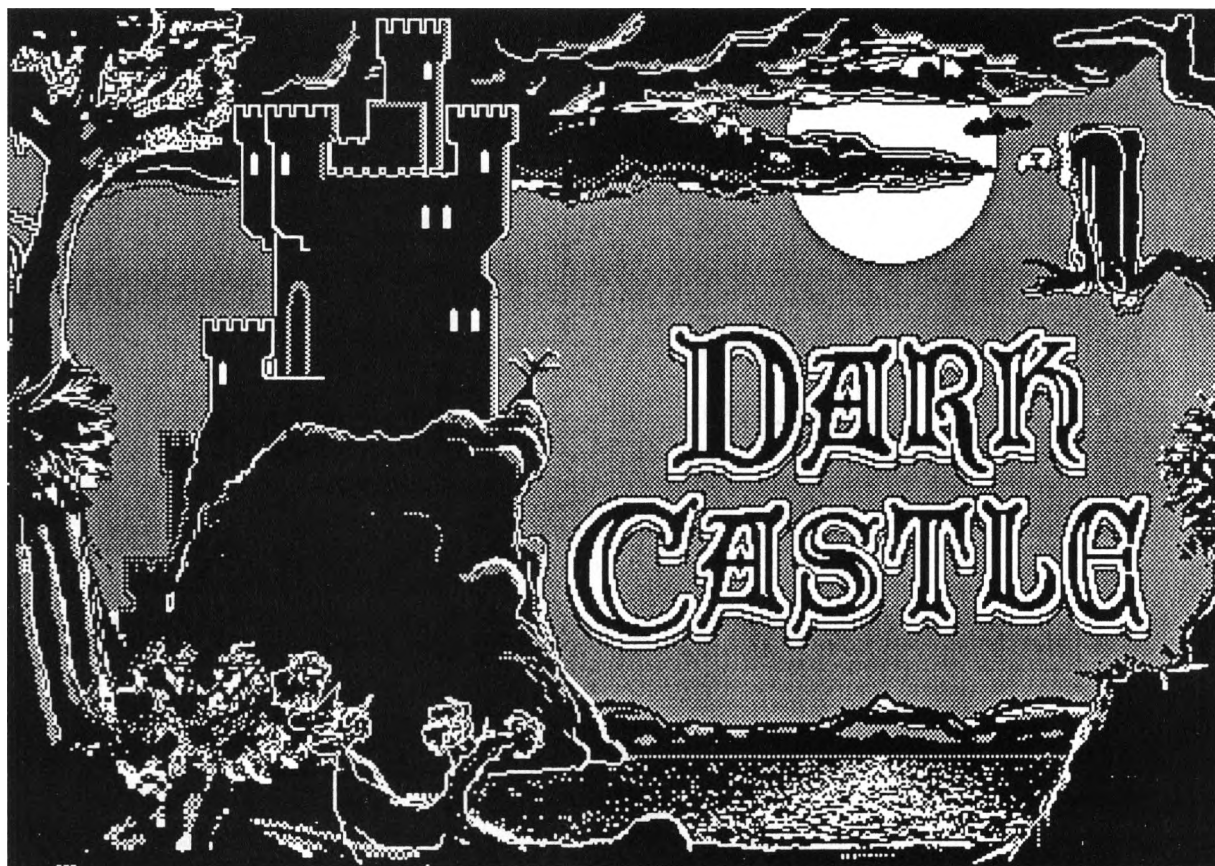
But why make so much of this distinction between font metric information and the lettershapes of a font? Simply because it is in fact not only a crucial logical distinction but also one that is central to understanding how word processing and formatting software works. All the real complexities of typesetting are inherent in the font metrics and only the rather trivial final stage of physically painting the final typeset pages involves the actual lettershapes. The $T_E X$ program, for example, knows absolutely nothing of letter shapes! Only screen previewing and "device driver" software that converts $T_E X$'s device independent output file into a bit mapped image knows or cares what a letter looks like. Of course in a WYSIWYG word processor, formatting a page and painting it on the screen are concurrent processes, so both the font metrics and the bit map images of the fonts must be known and used by these program, adding considerably to their complexity.



BMUG FALL 1986

Traditionally each type formatting system has had associated to it some unique method of its own for generating, storing, and using fonts. $T_E X$ for example has a highly sophisticated sister program, called *Metafont*, that is capable of creating for it fonts of remarkable beauty and flexibility. Using *Metafont* numerous font experts have collaborated with Knuth over a five year period to create a font family of great character (the Computer Modern or CM family) that, together with the "plain" format, makes $T_E X$ a complete typesetting system for basic typesetting.

To my mind one of the major advances made by the designers of the Macintosh family of computers was the idea of breaking away from this traditional mold and defining a standard Macintosh font format, thereby making fonts into universal and shareable system "resources", on a par with windows and menus. A user can create a font resource with a font editor and then install it in the System file, where it immediately becomes available to all Macintosh graphics and word processing application programs. This means that developers of Macintosh WP programs are relieved of the burdens of constantly re-inventing the font wheel and creating numerous fonts, thereby saving their efforts for more creative purposes.



Dark Castle screen shot



Editors Gordon Haig

File

I want to introduce anyone who is still using only *MacWrite* as their principal writing tool to the advantages to be gained from using an editor as a partner to *MacWrite*. Do your rough work in an editor which is simpler than *MacWrite* but faster and more flexible, then switch to *MacWrite* as your document nears completion or you need to work with a longer file. I have made extensive use of two editors, *File Utah* and *MEdit*. *File* is on BMUG Disk #3, and *MEdit* is on BMUG Disk #40 (the one that came with the Spring 1986 newsletter). They both give you an immense boost in speed and convenience of writing at a negligible cost.

File Utah is a text editor that was originally written to demonstrate the Mac interface. It is an ideal program for writing your rough drafts. It is small, only 20k; very very fast both loading and in operation; and allows you to have four windows open at one time.

The advantages of *File*'s speed are fairly obvious. *File* is always ready; you can pursue your train of thought without interruption. It doesn't make you wait while it completes some operation, never disrupts your rhythm of your work.

The advantages of multiple windows are perhaps less obvious if you have learned how to word process, as I did, using *MacWrite* as your principal application. However once you have experienced them you will not wish to do without. You can use one window as an on screen scrapbook, and save pieces of text out of the way but only a click distant. If you are rewriting or revising you can work in one window while in another window you can have an earlier draft, or an outline, or notes on changes to be made, or footnotes, or another section of the paper saved as a separate document. If you are working away and want to look at another file you don't have to close first; you just open an additional window. Or you could work on two or three sections of the paper at one time, switching back and forth. Cutting and pasting between two windows on screen is a snap; indeed with multiple windows it is easier to cut and paste from one document to another than it is to paste from one page to another distant page within a single document.

File supports three fonts only, is limited to files 32K or smaller in each window (say 16 pages) has limited formatting options and one significant bug. In the version with creation and modification dates of October 29, 1984, the **Save** command always saves to the disk from which *File* was launched, regardless of where the file started from. You can save to another disk only when you use **Save As...**, and only so long as you continue to use it. If you select **Save**, and the file was not opened from *File*'s launch disk, *File* will create a new file or overwrite an existing one of the same name in order to save to the launch disk. This last is the one dangerous aspect of this bug: since it is possible files of the same name on different disks, overwriting a file of the same name (without warning) may cause you to lose an earlier draft or to lose a set of notes you have saved under the same name. However ordinarily this bug is innocuous, and is easily allowed for.

When you wish to move from a draft to a formatted copy, a text file is easily opened in *MacWrite*, or opened in *MockWrite* and pasted into an existing *MacWrite* file. (Lovers of the arcane may wish to know that if you use the *Transfer* desk accessory to switch directly to *MacWrite* 2.2, a file left open in *File Utah* will come up already opened in *MacWrite*, for the utmost in fast transfers.)

BMUG FALL 1986

If you are using a Mac with the old ROMs and writing is one of the functions you use it for, I recommend this program very highly. It is small enough that you can conveniently keep it on the same disk as another application and system files, even with 400K disks. It can be used either as a partner to *MacWrite*, or to provide very considerable word processing power on a disk oriented to graphics or to another function.

Medit

Since I got the new ROMs, with the 800K internal disk drive, I have switched to *MEdit*, an editor which was included on the last BMUG membership disk. *MEdit* is similar to *File Utah*: it has speed, multiple windows, and small size. It is actually neither quite so very fast nor so very small as *File* (40K instead of 20) but it has other additional advantages that make it a delight to use.

MEdit can open multiple 32K windows, limited only by memory (and by screen space to display them effectively). In doing a recent project with seven sections, I created a file for each section, and had as many on screen at any one time as I needed, plus one for saving miscellaneous cuts and notes.

MEdit has sophisticated window handling mechanisms that let you use a large number of windows easily. For instance, one command allows you to tile the windows, and it will size and place all open windows (up to six) to be visible on the screen each as strips running side to side. If not enough of the document is visible, you can double-click anywhere in the header bar to expand a given window to full screen size, work in it, and when you wish to get to another window, just double click again and it will return to its original size, so that all the tiled windows are again visible.

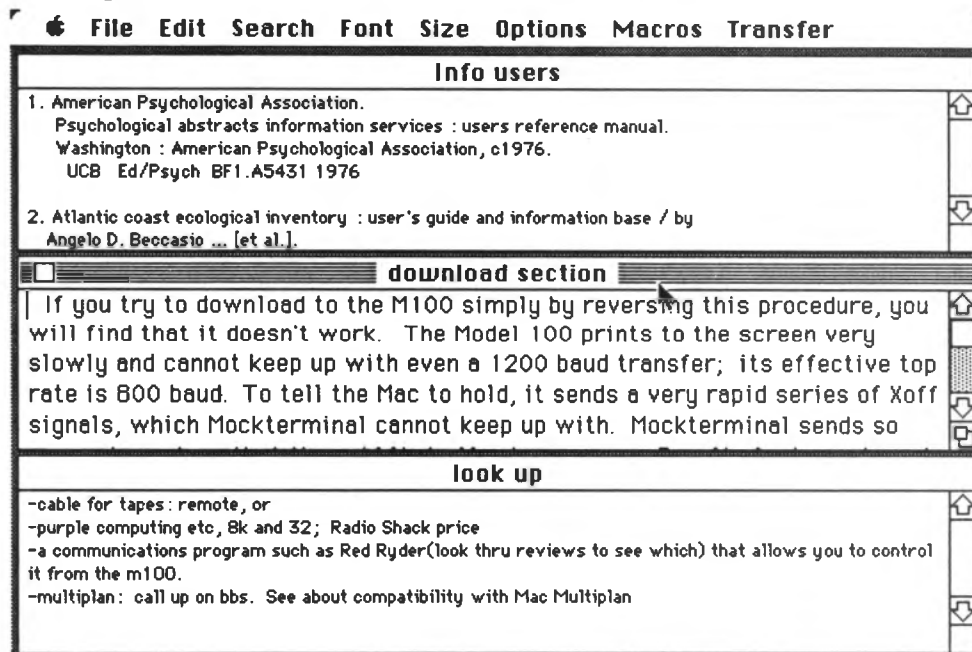


Fig. 1. Window arranged by *Medit*'s Tile command. Double clicking in header bar will temporarily expand window to fill whole screen.



If you have printed out multiple drafts and revisions of an article you are working on, as well as separate sheets with fragments or notes, it is often impossible to recognize, without reading all or part of each page, where each piece belongs. The capability of *MEdit* that won me over from *File Utah* is that with **Print Page Header...** checkmarked on, every time you print a file it is automatically given a header that includes your file name, date and time, and page number, and that is cleanly marked off from the main text by a page wide underscore. You don't need to do anything. If you write multiple drafts, each draft, and each section of the draft, is distinctively marked and immediately identifiable.

An additional advantage of *MEdit*, which I have never made use of, is that it has a macro language and you could potentially make templates for repeated headings, or boilerplate paragraphs, or macros for repeated operations. It comes with existing macro files for Pascal and for wordprocessing, and if you have more time than money, or specialized needs, macro files or templates for a variety of purposes could be created. There is also a transfer menu that makes it easy to go directly to *MacWrite* or another application from within *MEdit*. Although its printing and formatting abilities are primitive, *MEdit* is actually a very sophisticated application for the actual work of writing. The only commercial product with window management and template facilities nearly as sophisticated as this is *MORE*.

MEdit comes with documentation, in the form of an on disk file. However a brief word will make it easier to get started smoothly. The default settings in *MEdit* are intended for writing programs, not paragraphs. I have found the most convenient way to use *MEdit* is to create an empty document, and set the document parameters to suit your preferences. Then save the document as "*MEdit* Startup". If then you put the actual application in a folder, and start *MEdit* by double clicking on this document, it will come up with all your preferences set, ready to write. Just use **Save As...** to create a new file under the name you actually wish to use. Under the option menu I set the following: **Auto Indent**, **Line Wraparound**, and **Print Page Header** selected, the left print margin to 40 points, and then set the font to Geneva 12 point. The line length is fixed so the combination of left margin and font size effectively sets the right margin; with these settings your text is approximately centered on the page.

The text files *MEdit* create can be opened in *MacWrite*, in any application that accepts ASCII text files, in desk accessory editors such as *MockWrite*, or placed in the *Scrapbook*.

The multiple windows and sophisticated window management, the fast reaction time, the automatic name and time stamping of your files, and the easy transferability of text files into any other application make *MEdit* an ideal working tool for someone writing essays, articles, or multiple small documents. As your work nears completion you can load your files into *MacWrite* to work on the final form and formatting of your document.

Document Compare

Kenneth William Hirsch

Computerized word processing gives one enormous flexibility to revise documents in preparation. This flexibility to revise, revise, and yet again revise without having to re-type the whole document helps make contemporary report writing bearable. There is, however, a potential problem inherent in the production of multiple revisions — confusion over what is “in” or different in each version of the document. A partial solution to the problem is offered by the much longer document names which may be used for Mac documents than can be used on IBM-type personal computers. Most PCs allow document names containing only up to eight letters followed by a period and up to three additional letters. Eleven letters does not give one much to work with in trying to use a name as a clue to document content, especially two or three months — or years — later.

For example, I see in my disk catalog the names of two documents pertaining to an exchange of students between universities in Ireland and the U.S.A. which must have meant something to me at the time I created them, but now puzzle me. The documents were created on the Televideo computer I used to own, then converted to *MacWrite* documents by the utility *MacLink*, when I changed over to a Mac system. They are named EXCHUSAC.IR2 and EXCHUSAM.IR2. Without opening the documents and looking at them, I am stuck. When I do open them, you can bet I will make the names longer to tell me what is in each document!

There is a real possibility that these are simply different “generations” of the same document, with slight differences in wording. Such differences are often not apparent in a simple examination of first one document, followed by an examination of the other. Small changes often require concentrated reading and side-by-side, paragraph-by-paragraph comparison.

The program *Document Compare* finds and displays the differences. The Model Office Company, which markets it, needs to fix some bugs and add a function for saving a completed comparison on disk. (See below.) When it does so, I will give the program an unqualified recommendation. Meanwhile...

What It Does

Document Compare examines two documents side-by-side to show the differences, if any exist. The program is best understood by examining what it shows on screen. We shall see a comparison of two versions of the “Using... ThinkJet... Printers” article (in this issue of the *Newsletter*) to illustrate use of *Document Compare*.

Figure 1 shows the top-and-bottom layout of *Document Compare*’s selection dialog. Each of the documents to be compared must be selected from the two lists. Completion of selection activates the comparison process. After starting the comparison, *Document Compare* pauses and presents the screen shown in Figure 2, with ruler, picture, and word counts for the two documents being compared.

(Note: A problem exists in the program at this point. The manual states that the program works most effectively and quickly with text documents. However, when attempting to compare two multi-page documents, my copy of *Document Compare* gave a count and comparison of only the last of three pages.)

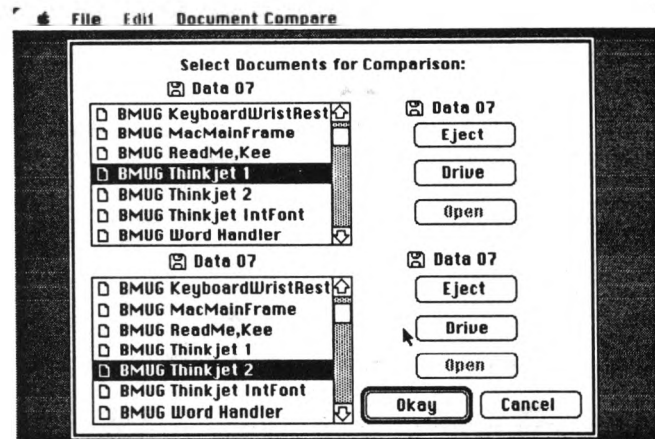


Figure 1. Document Selection Menu

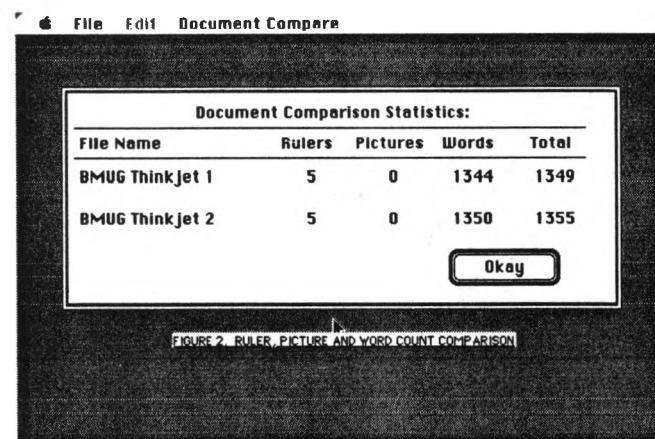


FIGURE 2. RULER, PICTURE AND WORD COUNT COMPARISON

In the next stage of comparison, the two document versions are scrolled past the screen, top half first. Each difference is reverse-highlighted as it goes by. Although it is possible to try to take note of the differences as they are presented my advice is to wait until the screen appears as shown in Figure 3. For long documents, you might like to go have a cup of coffee, for the process of comparison is rather like spell checking a document against an in-memory dictionary; it consumes a great deal of computer resources and requires considerable time. The manual gives suggestions for means to expedite the work, for example switching off a RAM cache while running a comparison.

After clicking on the **Okay** button, scroll each window to the beginning and work your way down, difference by difference, to the end of both documents, as shown in Figure 4. This screen should look the same as in figure 3, except without the dialog box.

A printout of the two documents, with their differences reverse-highlighted, is possible and very useful. However, long documents with few differences may be time consuming to print and wasteful of paper, not to mention requiring lots of space on the living room floor in order to look for the reverse-highlighted portions. The solution would be to save the comparison outcomes as a separate document, which could then be used with a word processing program such as *MacWrite*. One could then get rid of the large areas of the documents which are identical

BMUG FALL 1986

and re-save only those portions showing differences, along with markers such as page numbers. These, then, could be printed and be of great use in deciding which material to work with, perhaps alter yet again, and which to discard.

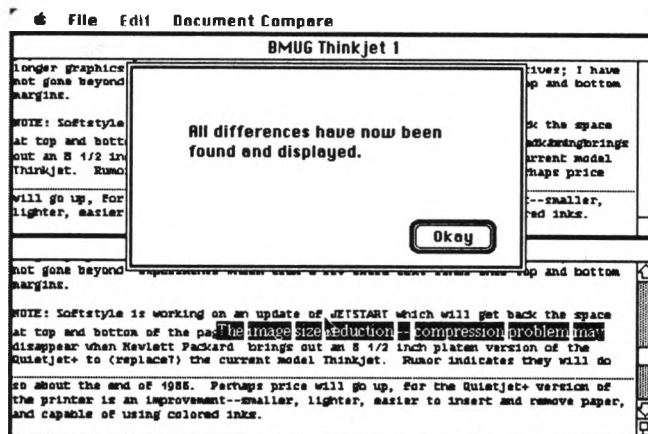


Figure 3. Completion of the Comparison

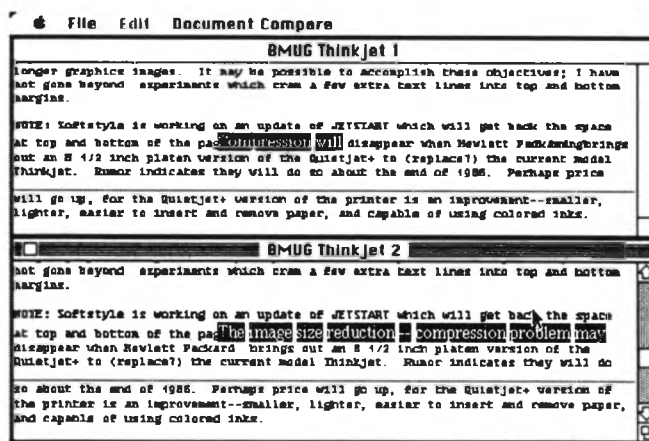


Figure 4. Document Differences

Ease Of Use

Document Compare is very easy to use and has a clearly written manual. There is on-screen help available which largely duplicates the manual. The program should be of great assistance for serious writing projects involving many revisions along the way.

Limitations

Document compare recognizes text (ASCII) documents, *MDS (Macintosh Development System)* Editor files, and version 4.5 or above ("Disk-based") *MacWrite* documents. There may be a problem using the program with documents nested into levels of folders under the Hierarchical File System (HFS). If you are using the HFS with nested folders, you should check out the program first to see if it will work. Model Office Company does offer a 60 day money back guarantee. Still, caveat emptor.



Microsoft Word documents must first be saved as text documents, then compared. The same must be done if one wishes to compare spread sheet data, data base information, and other forms of data savable as documents.

Although there are several options specifiable from the menus, the default settings should be appropriate for almost all uses. Nevertheless, for each re-start or new comparison, the program always re-shows the application disk menus, even though the rather time consuming shift to the disk on which one's documents are stored has been performed for every previous comparison.

There is no batch mode of operation, in which one could click a series of documents to be compared, one after the other, then either saved to disk as separate documents or printed onto paper.

The absence of the capability to save the comparison outcome to a separate document is a serious limitation, as it prevents using any printer other than ImageWriter or LaserWriter. To print the outcome on a formed character ("daisy wheel") printer usually requires operation of a separate program (a "printer driver" program such as *ProPrint*) upon a separately named and saved document.

Formed character printers are very common in business and university environments; they offer advantages over dot matrix printers such as ImageWriter when used to print non-graphic documents. They may not be as flexible as a LaserWriter, but then they have the great advantage of not costing five grand. Peter Hart, the owner of The Model Office Company, claims a newer version of *Document Compare* will incorporate a save-to-disk function.

Problems

Document Compare wouldn't run when it came to me from Canada. The Company claimed a series of "bad disks" had been sold to them— sorry, etc. (Don't they ever try out what they sell?)

The second disk they sent ran—but was equipped with obsolete *Finder* 4.1 and its associated obsolete *System*. I believe this was because the company is not confident of the program's ability to operate correctly with documents inside nested folders within the Hierarchical File System provided with current Macs.

Document Compare was copy protected in a manner I couldn't solve using *Copy II Mac* during two wasted days trying to get the program onto an 800K disk, married to current *Finder /System* 5.3/3.2.

The third disk they sent—as before, after my phone calls, — arrived without copy protection, ran, and thus finally permitted me to make the installation of *Jetstart*, necessary to get a printout of the results via my Hewlett Packard ThinkJet printer. 'Till then, the program had not been of real use. Accompanying the disk, however, was an astounding statement in an otherwise pleasant letter, in which the company asserted it was sending me the uncopy-protected disk because they knew that I was honest. How do they know that? Is it because I phoned and bitched enough to cause them to respond? Because I am a university professor? Because my name, Hirsch, is not too difficult to spell? Because I live in California? Because I have a nice voice? What would they conclude if my name was Kraekenpfafferr? If I had a gravelly voice? etc.? This seemed especially ironic in view of the irritating and redundant quasi-legal notices they plaster on the manual and program. A portion of my response letter to Peter Hart may be of interest:



BMUG FALL 1986

"One final note: After having signed your licensing agreements, having read them again on-screen (while instead expecting to see program use information), and having noticed their redundancy and profusion with the associated implication of expected dishonesty, I conclude that many — perhaps most — of your customers will become resentful. That is, after reading and signing on paper, then reading again on disk AND THEN ENCOUNTERING A LOCKED-IN, COPY PROTECTED PROGRAM WHICH REQUIRES THEM TO GET HELP FROM SOMEONE WHO KNOWS HOW TO REMOVE THE COPY PROTECTION, I would bet their initial commitment not to give away your programs so as to help you sell them would be lessened, not enhanced."

Recommendation

This is a special purpose program, with a great deal of potential benefit to writing projects. It is expensive — \$99 in U.S.A. currency — (they accept checks on U.S. banks) via mail to:

The Model Office Company
The Gooderham Flatiron Building
49 Wellington Street East
Toronto
Canada M5E 1C9

If you can do without it for a while, I do not recommend purchase of *Document Compare* until: the program incorporates a save-to-disk ("print-to-disk") function for the comparison outcomes, copy protection is removed, the program is made to work fully with hierarchical file system, and the multi-page text (ASCII) document problem is fixed.

About the Author

Kenneth William Hirsch was recently confused with being a medical doctor, which he is not. (See "Word Handler".) He is a Professor of Communication Studies at California State University, Sacramento, where he is currently convening a new Sacramento University Macintosh group (SUMAC?), after having done much the same thing to help start the Sacramento Osborne Group (SOG) more than four years ago. SOG is still going strong. Hirsch makes no such claim for himself.

Msg. #2030 in **Public Messages** Posted on 07/08/86 at 11:06:51
To: ALL From: MICHAEL BEAUCAGE
Subject: Large screen for Mac
Concerning Thomas Gilley and his inquiry about a large screen for the Mac,
I have been using a COMTREX modified Electrohome 19" monitor and patch
board for over a year and it is great! People who see it for the first
time say "that is what the Mac screen should look like", very Sun-like.
call Steve
Baker at (714) 675-8608 in San Diego or you may call me at (415)
524-9933. Cost is about \$1000.00.
Good investment if you work on the Mac with other people



WYSIWYG Technical Word Processing

Richard S. Palais

WYSIWYG technical word processing (TWP) on the Macintosh is a paradox. In contrast to the chaotic profusion of WYSIWYG word processors for the IBM-PC, many with excellent TWP features, *the Macintosh world currently does not have even one WYSIWYG technical word processor*. Nevertheless mathematicians seem to find it remarkably easy to get excellent results writing up their research on the Mac. The explanation of this apparent contradiction can be found in several nice features of the Macintosh:

- The intuitive and uniform user interface makes it easy for even new users to learn and to integrate a lot of software with very little effort. This contrasts with traditional systems in which learning to use a new program, even one very similar to another in which you are expert, can be a long and tedious process of memorizing one more arbitrary command syntax.
- Most computer operating systems only permit a program to act on the output of a *previously* run program (the UNIX pipes mechanism is a powerful formation of this concept). But, in the Macintosh system, at least two mechanisms allow well constructed programs to operate as co-processes on the same data, interacting across the standard Clipboard and Scrapbook interfaces. These of course are *Switcher* and Desk Accessories.
- Macintosh fonts are highly flexible, editable, shared resources. As we shall see, the ability to create and modify fonts at will can be a powerful tool for overcoming limitations of word processing software!
- There are several excellent macro processors for the Macintosh, (*MacTracks* and *Tempo* being two of the more popular ones).

Taking advantage of these features mathematicians have been able to collect many different pieces of software off the shelf and fashion them into a toolkit to create their own WYSIWYG TWP environment. This makes it difficult to review software in the traditional way. Each program must be considered not on its own, but rather on the basis of how it fits in with other software, often software it was not explicitly designed to work with. And there are many different systems that can be fashioned in this way, while I could hope to tell you (or even know) about only a few of them. In fact though, many people have written to me explaining how they overcame one or another limitation of the basic WYSIWYG word processors, and whereas there is a lot of diversity in detail, a definite pattern or consensus seems to be emerging, and I shall to try to explain roughly what I feel this is.

Let's consider what features a technical word processor should have. First, it needs the basic elements of a word processor designed to handle just ordinary, non-technical text (e.g., insert and delete, find and replace, block copy and move, the ability to move easily between different parts of a document, text justification, the ability to set page numbers (or folios), headers, and footers, margin setting, tab setting, the ability to change fonts and font sizes). *MacWrite* is reasonably satisfactory on this score and I will assume all readers are familiar with it. It does lack the ability to simultaneously open several files (or several windows into the same file), and it also lacks such niceties as style sheets, flexible control of leading, word abbreviation, mail merge, and automatic footnoting. These features, lacking in *MacWrite*, do exist in *Microsoft Word (Word)*, but are not particularly necessary for the sort of work most mathematicians do. So

it is not too important whether you use *MacWrite* or *Word* as your basic word processor. (*Word* by the way does have a couple of disadvantages compared to *MacWrite*; it is copy protected and it is not completely WYSIWYG). Two other WYSIWYG word processors for the Mac should be available by the time you read this. One is called *MacAuthor* and is development by Icon Technology Ltd. of Leicester, England. (See Peter Korn's review elsewhere in this issue.) The early versions I tested were both slow and exceedingly buggy, but apparently it is gradually being improved. I just received version 1.2, which seems to be faster, and it is also supposed to cure many of the bugs in earlier versions, but I have not had time to test it extensively. *MacAuthor* has many nice extra features, like kerning, but there is one in particular which (see below) can solve many of the problems of TWP, namely non-destructive backspacing. On the other hand they have made the process of changing fonts and font sizes so complicated that this alone may kill it as an option for TWP. The second is *WriteNow* (née *MacAuthor*), created by Steve Jobs' new company, NeXT Inc., and to be marketed by T/Maker. I have not had a chance to test it personally, and saw it first only a few days ago at the MacWorld Expo. It was being demonstrated by its programmer, D. John Anderson, who told me that the original specifications called for *WriteNow* also to have non-destructive backspacing. But, sad to say this feature was dropped somewhere along the way. It seems beautifully crafted, fast, and with many nice features. But what I think may prove most useful for TWP is that in *WriteNow* an imported graphic is treated as just another character box, so that a formula created by an equation processor can be inserted in-line, rather than having to be displayed. (See John Heckendorn's review elsewhere in this newsletter.)

But now let's concentrate on the *technical* in technical word processor. What characterizes mathematical or technical text, as distinguished from ordinary text, is that it contains lots of formulas (or equations) and diagrams. And so what distinguishes a technical word processing system from an ordinary one is the ability to handle these extra elements easily. Since making long vertical arrows is difficult within a word processor, and slanted ones even more so, diagrams of the commutative type are usually prepared in *MacDraw* or *MacPaint* and then transferred either directly to *MacWrite* via the Clipboard (this is a lot faster and easier using *Switcher*), or first put in the Scrapbook and later transferred to the *MacWrite* document. Other types of pictures and diagrams are also easily handled in this way. But this is definitely *not* a satisfactory way to handle formulas. Lets see why.

When one first thinks of how it might be convenient to handle equations in a WYSIWYG environment it seems that all that is needed is the possibility of moving the cursor around freely to an arbitrary location on the screen and then entering an arbitrary symbol from an arbitrary font at that point. But mathematical formulas have a rich, recursive structure, and while it would be possible to ignore that structure and paint formulas on the screen in this haphazard way, the resulting system would be very difficult and tedious to use and, unless used with extreme care it would give ugly and uneven results. If you are taking the square root of an expression, for example, the size of the radical sign depends on the size of the subformula for the expression, and editing the subformula can change that size. Similarly when creating a matrix or other tabular array it is difficult to align the array elements and to have the enclosing brackets come out the correct size and in the right place. Anyone who has tried to use *MacWrite* or *MacDraw* to create complex formulas or arrays in this way knows how difficult it is to get acceptable and uniform results. Fortunately several programmers have created equation processors. These have built into them a great deal of knowledge about the structure of mathematical formulas and take much of the frustration and tedium out of the formatting of equations. Their use is based on an intuitive stack based discipline. For example if you tell the system you want to make a fraction, the cursor first jumps up to the numerator, and after you type it in and type **Enter** the cursor next jumps down for the denominator, and when you finish this and again type **Enter** the



equation processor centers the numerator and denominator and puts the correct length solidus between them. Similarly, if you ask for an integral, the integral sign is drawn and then the cursor jumps first to the lower limit, and when that is finished to the upper limit, and finally to the integrand. Of course one of the limits of the integral might be a fraction, the denominator of which could be another fraction, etc. One quickly gets the hang of using an equation processor, after which creating formulas becomes easy and almost enjoyable, and it is hard to imagine ever doing without one again.

At the moment there is only one fully operational, released equation processor for the Macintosh. This is *MacΣqn*, created by Dennis Venable and available from Software for Recognition Technologies, 110 University Park, Rochester, NY 14620 for \$44.95. Venable has configured *MacΣqn* as a Desk Accessory, making it very convenient to use, but of course DA's are more limited in size than ordinary applications, and he has shown great programming skill in squeezing so much functionality into 18K! Early versions of *MacΣqn* had two annoying defects: you could not edit an equation after it was created, and you had access only to the four LaserWriter fonts. I have just received the latest version which fixes both of these problems. (See Doug Shier's review elsewhere in this newsletter)

There are two more equation processors I know of under development. One, called *Weinberg*, is an ambitious project by Allan Bonadio. The version I have used looks promising, but is still in too primitive a state for me to guess how successful it will eventually be.

The other one, programmed by Ted Sobel, is quite polished and is nearly ready for release. Called *MathWriter* will be sold by Cooke Publications, P.O. Box 4448, Ithaca N.Y. 14852, for \$50. (By the way, similar programs for the IBM-PC sell for five times as much and more!). I only received my review copy a few days ago at the MacWorld Expo, but I have spent a couple of hours playing with it and have been very impressed with both the program and its documentation. The designers have chosen an almost opposite design philosophy from that of *MacΣqn*. If *MacΣqn* is a lean, mean racing car of a DA, then *MathWriter* is by contrast a luxurious sedan of an application. The program is loaded with features that make it easy to use, but they also add to its size, which is a very hefty 115K (for comparison *MacWrite* is only 70K). Since it needs about 250K of RAM to run, and *MacWrite* needs a similar amount it seems that to run the two together under *Switcher* you will need a Mac Plus.

Now the bad news! Ideally a technical word processor would have an equation processor built into it, and formulas would be created in place. But in today's Macintosh world, equation processors are implemented either as a Desk Accessory or as a separate application. This means that after a formula has been created in an equation processor it must then be transferred to the Clipboard in one of the public data formats (e.g., as a Pict file) and then pasted back into the word processing document. And this causes a problem! Both *MacWrite* and *Word* consider an imported graphics file to be (in T_EX terms) a vertical box, i.e., analogous to one or more lines of text, rather than a horizontal box, like a character, that can be inserted into a line of text. This means that an imported formula *cannot* be pasted into a line of text as an in-line formula, but *must* be pasted between lines of text, i.e., it must be a displayed formula. (However, see the above remark concerning the forthcoming *WriteNow*).

It might seem then that equation processors have *not* solved our basic problem, namely creating formulas with a non-technical word processor, since the great majority of formulas in mathematical writing are in-line, and these must still be handled by the word processor itself. But this reduction of the problem, to one of handling only in-line formulas, turns out to be the key to a simple solution. If you examine a large sample of mathematical text with this *in-line*

versus displayed distinction in mind, you will soon detect clear patterns in how mathematicians and typesetters decide whether or not to display a formula. These decisions are apparently based on intuitive rules that we all seem unconsciously to know and to use. In-line formulas are generally short and simple. Most in fact are only a few symbols long, serving merely to name or refer to some objects or to state some simple relation (equality, membership, or inclusion) between two typographically simple objects. Without going into further detail, let me say that a word processor will be adequate for handling in-line formulas if it can just do the following, quickly and easily:

- Switch to any font or font size.
- Attach to characters on the baseline both sub- and superscripts, *with the possibility of a superscript over a subscript*. Moreover, these first order sub- and superscripts should themselves admit sub- or superscripting. Finally the baseline characters and at least the first order sub- and superscripts should admit decoration by accents both above and to the side.

For example: it should be easily possible to include the

formula $\bar{g}_k^a$ within a line of text. Now while the latter

formula is very easy in $T_E X$, and is not hard with an equation processor (I used *MathWriter*), it is enough to make a beginning *MacWrite* user give up TWP in disgust.

The problem with *MacWrite*, of course is that, after you insert a character, the insertion point moves forward by the width of the character box so, for example, after typing a superscript there is no way to place a subscript just below it. If you hit the Backspace key, then of course the insertion point will move back, but the superscript you just inserted will simultaneously disappear. In other words, the Backspace key actuates a *delete and backspace* or *destructive backspace* operation. In *MacAuthor* though, if you hit **Option Backspace** after typing a character, then the insertion point backs up to where it was before that character was typed *without erasing the character*; this is what is called a non-destructive backspace. Using it you can overstrike characters, for example creating a $\neq$ symbol by overstriking an equal sign with a slash, but much more importantly, if you do this trick after typing a superscript you can then type a subscript just below it! Similarly non-destructive backspacing makes it easy to create simple in-line fractions.

Fortunately there is a trick for simulating the effect of non-destructive backspacing in any Macintosh WYSIWYG word processor. If you recall the above discussion of fonts, particularly the point about characters not having to be inside their boxes, you can probably guess how it is done; just make a character's box have a width of zero, and let the character jut entirely out to the left side of its box. When you type such a character the insertion point will stay put, *just as if you had followed it with a non-destructive backspace*. If you modify an entire font so that all of its characters have zero width, you will in effect be able to do a non-destructive backspace after inserting any character from that font. But recall that the major reason we wanted non-destructive backspacing in the first place was to be able to put one character over another. We can make this even easier if we create not one but *two* zero width versions of the font, one (for superscripts) with the characters shifted up from the baseline by seven points and one (for subscripts) with the characters shifted down by five points. Finally I would recommend that you do this with a nine point version of the same font that you use as your main text font at twelve points.



If you do this with the Boston font (version II) you will have exactly the the two fonts I have in my System under the names Up and Down. With these fonts and a little experimenting you will soon discover that it has suddenly become surprisingly easy not only to put superscripts over subscripts, but also to make fractions and two levels of sub- and superscripts. A little warning though; you probably should not make your fonts really zero width. Certain versions of the ImageWriter driver can not handle zero width characters, so although your text might look just right on the screen, if you have one of these versions of the ImageWriter driver in your System Folder you may find that zero width characters will not print! To be safe you can make the characters all one pixel wide instead. (Of course, to be able to do high quality printing on your ImageWriter you should also make two pixel wide, eighteen point versions of these fonts). If you have the font editor *Fontastic* from Altsys, and a little experience in using it, then carrying out this program of font modification should be about an evening's work. But there is a better and easier option. Bill Fulton and Joe Harris of the Mathematics Department at Brown University have created by far the best set of fonts for WYSIWYG mathematical word processing on the Mac I have yet seen. It will be sent to you, along with a short but clear manual, if you make out an eight dollar check payable to Maine Fonts and mail it to: Maine Fonts, Dept. of Mathematics, Box 1917, Providence RI 02912. The Maine family has as its principal text font a font named Machias that has borrowed the marvelous roman lettershapes from Charles Maurer's BostonII font.

But for mathematical text the choice of extra characters in Machias is a great improvement over Boston. Maine also has an excellent new symbol font called Lovell, and it implements (in a slightly different way) the zero width, raise and lower trick mentioned above. Fulton and Harris are considering making a LaserWriter version of their fonts if there is enough interest. By the way, the only WYSIWYG LaserWriter technical fonts that I know of come as part of the new Version 3 of the Princeton Fonts of Dougherty and Lam, available from many user groups, or by writing to: Princeton Font, Department of Mechanical Engineering, Princeton NJ 08544 (suggested contribution is \$30). Princeton was the first TWP font for the Mac, and probably more mathematical papers have been written on the Macintosh with it than with any other font. (It was also the first font that I know of to use zero width characters, although in a much more limited way than that suggested above). My only reservation about it is that it is based on the sans-serif Geneva font, which I find somewhat illegible and lacking in character.

There are many TWP fonts advertised and available commercially. I've tried a number of them and my feeling is that for the most part they were created following someone's theoretical idea of what technical writers *should* need rather than studying what they in fact do use. They are generally serviceable but there are none I have tried that I can recommend with any enthusiasm. Like any fine tool, a good font family has usually evolved gradually from the experience and needs of its users. I know this to be the case with the Maine and the Princeton fonts and I am sure that there are many more such fonts of this same high quality created by working mathematicians to fill there own needs. Let's hope some method can be found to publicize and distribute such fonts. They are a precious resource!

An Equation Maker for the Mac

Doug Shier

Technical word processing is becoming increasingly important in the mathematical, physical, and engineering sciences as more and more authors (students and faculty alike) are attracted by the charms of the microcomputer. Since the visual feedback provided by a "what you see is what you get" environment is especially important for technical word processing, the Macintosh interface seems a natural candidate for this task. Unfortunately, even the fairly simple demands of technical typing (such as multiple levels of subscripts and/or superscripts) quickly reveal the limitations of current word processors for the Macintosh. To those of us who habitually lapse into the mire of complex equations, the only real alternative has been to painstakingly design such expressions in *MacDraw* and then transfer them to the required document. While the inherent virtue of constructing by hand these intricate designs cannot be denied (even equations can be beautiful), many would agree to be tempted from the righteous path by the right software. If so, then *MacΣqn*, an "equation processor" for the Macintosh, offers a seductive alternative.

MacΣqn is a desk accessory that can be invoked from your favorite word processor (such as *MacWrite* or *Word*) or from the desktop itself. This DA is of modest size (about 19K), runs on both 128K and 512K machines, and works without difficulties in conjunction with the new System 3.2 and HFS. The object created within *MacΣqn*'s movable window is stored in PICT format and is transferred to the current document via the clipboard, just as an object created in *MacDraw*. *MacΣqn* allows the construction of complex expressions involving multiple levels of subscripts and superscripts, integrals, summations, products, fractions, square roots, matrices, and devious combinations thereof. In addition, it is possible to place bars, tildes, and hats over any symbol, as well as bars over any given expression, however complicated. The constructed expression prints out exactly as displayed, and certain (limited) editing features, described below, are also available. It should be noted that *MacΣqn* works best with the LaserWriter fonts (Helvetica, Times, Courier, and Symbol), but other fonts can be accommodated. Results can be printed out on either an ImageWriter or a LaserWriter. At any point in the process of constructing an expression, various fonts, sizes, and styles are available for selection.

What It Does

Since a picture is worth at least a few K words (and equations are just someone's picture of the world), the following five sample displays ought to illustrate what can be done - and done rapidly - with this package. The first such example is taken from a collection of benchmarks for mathematical word processing packages and it was created in only a minute or two using *MacΣqn*.

$$W_{m_1 n_1 n_2}^{3\beta}(p_1, p_2) = U_{m_1 n_1}^{3\beta}(p_1, p_2) + \int_0^\infty \frac{dp_3 p_3^2}{8\pi^3} \sum_n \sum_m \sum_{\alpha_2} \sum_{\beta_2} \sum_{n'} \sum_{n''} (-1)^m$$

$$\frac{\sqrt{x^2 - y^2}}{x^2 + y^2} = g(x, y)$$



$$\hat{H} = -\frac{\hbar^2}{2m} \nabla^2 + V(r)$$

$$P(\Theta_1 | x) = \left(\frac{\rho}{2\pi}\right)^{1/2} \int_{-\infty}^{\theta_0} \exp\left\{-\frac{\rho(\theta - \mu(x))^2}{2}\right\} d\theta$$

$$\nabla f = \left[\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \cdots \quad \frac{\partial f}{\partial x_n} \right]$$

How It Operates

MacΣqn employs a stack to keep track of the state of the current expression. For example, by entering a Command-S (or selecting SUM from the *MacΣqn* menu), a summation sign is displayed in the window, and space is allocated on the stack for both a lower limit and an upper limit. The user then enters a subexpression for the lower limit (which itself can be complex), followed by a subexpression for the upper limit. The special ENTER key is used to signify the completion of a particular subexpression. This way of processing and structuring the input turns out to be quite natural. Moreover, it is exceedingly powerful. Using this technique, complex expressions can be built up layer by layer from simple ones. In addition to the visual Mac interface (that shows the status of the current expression), an information line present at the bottom of the *MacΣqn* window is helpful in navigating through the stack. Once having seen how this stack operates, it becomes clear that this is the "right" way to construct a variety of complex expressions using only a few simple building blocks.

A Few Features

A number of nice features have been built into the package. For example, after entering a summation sign, the lower and upper limits are automatically resized to a smaller point size. (A similar downsizing occurs when using integrals or products, and when subscripting or superscripting variables. This feature can however be overridden.) As an added feature, the limits are automatically centered relative to the summation (or integral, product) sign. In addition, some fine tuning in position can be accomplished by selecting a given symbol and using cursor keys (from a numeric keypad) or keyboard strokes to move a pixel at a time in any direction. Character replacement is also supported. More extensive editing can be performed by transferring the *MacΣqn* expression to *MacDraw*. There are also other goodies described in the well-written manual, including a description of how to customize the package to accommodate different fonts and sizes. It is apparent that the developer gave considerable thought to not only what users might want to do but also how to achieve this easily and painlessly.

A Few Quirks

First, it should be emphasized that this desk accessory creates displayed, rather than in-line, equations. Like any *MacDraw* object, a *MacΣqn* expression stays where it has been placed and text does not flow around it when using *MacWrite* or *Word*. It is possible though to create the illusion of continuous text by inserting a *MacΣqn* line into the middle of a paragraph of text. This might be desirable for example when an overbarred or hatted symbol needs to appear or



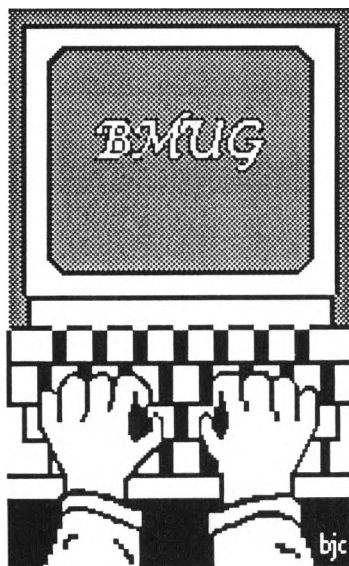
BMUG FALL 1986

when a simultaneously subscripted and superscripted variable appears. However, this insertion disturbs the usual even spacing of the paragraph. For 1-1/2 line paragraph spacing, the judicious insertion and manipulation of rulers in *MacΣqn* can create a seamless body of text. Just place a single-spaced ruler immediately before the *MacΣqn* display and a 1-1/2 spaced ruler after the *next* line following the display. This same sort of trickery by ruler is less satisfactory for double-spaced material, unless one uses *Word* — in which case one has complete control over the spacing, down to the very last point of size. With the availability of newer word processors (such as *Works* and *WriteNow*), which allow the insertion of PICT objects into the *middle* of a line, much of this fiddling around will become unnecessary.

The first release of *MacΣqn* (November 1985) has a few idiosyncracies that should be noted. For example, cutting from *MacΣqn* and pasting to *MacΣqn* are not supported in this release. Certain editing operations (Copy, for example) should only be selected with the mouse we are told and not from the keyboard — else the system may crash. Also, the refresh command (which is used to redraw a clean display) can at times change the point size from the value set before the refresh occurred. Finally, a *MacΣqn* creation pasted into *MacWrite* cannot be properly aligned with the (default) left margin, set at 1-1/8 inches. To achieve exact alignment, new rulers can be inserted immediately before and after the *MacΣqn* construct and the left margin readjusted slightly to the left. Fortunately, all of these problems appear to have been resolved in the latest release (Version 1.9, August 1986).

MacΣqn is certainly not a substitute for a good technical word processing system. However, until that glorious day when such a package arrives, anyone who communicates with equations should find a comfortable companion in this application. Priced at a reasonable level, packed with a lot of power, and not burdened with elaborate copy protection schemes, this is the type of software we should all demand and support.

MacΣqn from Software for Recognition Technologies, 110 University Park, Rochester NY 14620.



Processing Words/Chan



T_EX for the Macintosh from K&S and FTL

Richard S. Palais

The competition between *Personal T_EX* and *PC T_EX* for the IBM-PC T_EX market has not only meant lower prices, it has also spurred continued improvements in each. Now it seems that the Macintosh world will have the same good fortune. Two implementations of T_EX for the Mac have been announced. One created by Kellerman & Smith of Portland, Oregon is being marketed by Addison-Wesley for \$495. The other is from FTL Systems Inc. of 234 Eglinton Ave., Toronto, Ontario, Canada M4P 1K5, and is priced at \$795. Since both have chosen *MacT_EX* as their trademark, to avoid confusion I will refer to them as *K&S T_EX* and *FTL T_EX*.

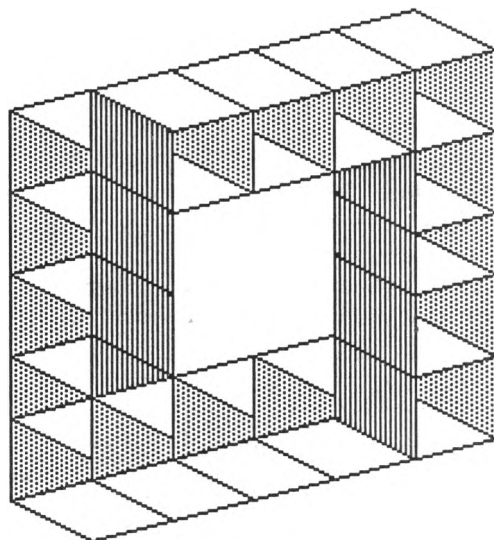
Both versions are full implementations of T_EX, of course. In particular they have passed the suite of tests Knuth has devised for systems that want to call themselves T_EX. In fact, as far as functionality goes, both systems have taken advantage of the Macintosh environment to go well beyond T_EX in rather similar ways. Both have a standard Macintosh editing environment built in for creating and editing T_EX source, both have screen previewers and have provisions for pre-loading macros and for interspersing *MacPaint* and *MacDraw* graphics with text. Both support the ImageWriter for proofing, but in addition they support the LaserWriter fonts, have AppleTalk drivers, and permit inclusion of PostScript commands along with T_EX source, so that final camera copy can be output not only on the LaserWriter, but also on much higher resolution PostScript phototypesetters such as the Linotronic 100 or 300.

Having said all this, it might seem that the two *MacT_EX*s must be as alike as two peas from the same pod. But in fact K&S and FTL clearly had *very* different user bases in mind when they created their implementations, and this is reflected in user interfaces, which have a very different feel. K&S created a version of T_EX for a fairly naïve user who wants to be spared the innards of T_EX and wants to get his output with the least fuss possible. If you are a Macintosh user familiar with the standard way Mac programs work and have had a little experience using T_EX on a mainframe, then you can slip two disks in the drives of a Mac Plus and get down to work without ever looking at a manual. Writing the T_EX source, running it through T_EX, previewing it on the screen, printing it on the ImageWriter (or on a LaserWriter if one is attached) are all so completely straightforward and natural that it took me some time to appreciate the remarkable planning that made this simplicity possible. Aside from the File and Edit menus there are only three others, a Typeset menu (you pull it down and click **Typeset** when you are ready), a View menu that allows you to choose the page of a typeset document that you want to preview and the magnification you want to see it at (including one magnification that fits the whole page in the preview window), and a Windows menu that lets you switch easily between several different T_EX source files, their preview windows, and their associated graphics Scrapbooks. Unless you are doing something *very* fancy, the only point which needs a little explanation is how to include graphics. This is handled very neatly, in a standard Macintosh way. Each T_EX source file has associated to it a Scrapbook. Using the Clipboard as usual you can put any *MacPaint* picture or *MacDraw* diagram into this Scrapbook and give it a name, for example MyPicture. Then essentially all you have to do is place the macro `\picture{MyPicture}` where you want it to appear.

FTL T_EX, on the other hand, feels very much like an implementation for a T_EX hacker. It has been aimed for example at a T_EXpert working for a publishing house. It has three more menus than K&S, and the menus are more complex, with more options. It is *not* the kind of program to use without a manual, and fortunately the manual is quite well written. (It is not without its faults, however; on page four the manual starts out by saying that to run the program you *will*

need a Macintosh Plus, a 20Mbyte hard disk, a LaserWriter Plus, and an ImageWriter with Thunderscanner. This is of course silly, and two pages later you are informed that you can in fact fit a minimal version of *FTL T_EX* on a single 800K disk). All kinds of things are in fact a little or a lot harder with the *FTL* version. You cannot, for example, start it up by double clicking on a document it has created. And if after previewing your file and checking that all is well, you select **Print** from the File menu with the LaserWriter chosen you get a nasty surprise. The top margin is off the top of the LaserWriter page; it seems that *T_EX* and the LaserWriter have different ideas about where the origin of a page is and some special incantation, not mentioned in the manual, is necessary to make the printing come out right—not exactly user friendly. Worse still is the complex method used for getting graphics into a *T_EX* file. It essentially forces the user to do some PostScript programming. Something else that I feel is a defect in the *FTL* version is the primacy given to the LaserWriter fonts over the usual *Metafont* created fonts that usually go along with *T_EX*. Not only are the LaserWriter fonts the default for the Plain macro package, but one cannot even use the screen previewer with the native *T_EX* fonts! And the *T_EX* fonts that are supplied are the old, preliminary AM series of fonts, rather than the much improved CM family. But one should not lose sight of the fact that this is a very solid program, and what faults it has can easily be fixed, and no doubt will be in response to its very strong competition. By the way, one way in which I feel *FTL T_EX* is better than *K&S T_EX* is in its handling of *T_EX* error messages.

I did a comparative timing test for the two implementations. The test document was four pages of fairly dense mathematics and I ran it on a Mac Plus with an external 800K disk drive and output to a LaserWriter Plus. *K&S T_EX* took one minute and twenty-five seconds to do the typesetting, as compared with three minutes and five seconds for *FTL*. Previewing (with whole page in the window magnification) took thirty seconds for *K&S* as against four minutes and fifteen seconds for *FTL*, and printing on the LaserWriter took one minute and forty-two seconds for *K&S* and three minutes and thirty-five seconds for *FTL*. By the way, when I ran the same test at home on my Levco Monster Mac, running out of a 1Mbyte RAM disk, *K&S* took only forty-nine seconds to typeset and fourteen seconds to preview.

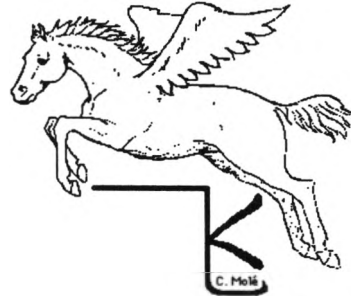


· Boxes/Hockabout

BMUG
Fall 1986



FeiMa Chinese Language Processor Barbara J. Chan



Introduction

Mr. Gene Wu, head of Wu Corporation and chief designer of the *FeiMa* Chinese Language Processor, demonstrated *FeiMa*, a Chinese Language Processor, and explained Wu Corporation's philosophy about computers serving the needs of users.

Unlike Chinese word processors available on other computers such as the IBM PC, *FeiMa* does not require the user to learn a new programming language, or a complicated interface system. Wu Corporation chose the Macintosh computer because it allows users to concentrate on creating and producing work, and does not require users to learn complicated keyboard commands. *FeiMa* was designed with several key factors in mind: to be an easy language processing program for speakers and writers of Mandarin Chinese which would not require them to know English, and to be a learning tool for students of Mandarin. *FeiMa* includes both "Fantizi" and "Jiantizi" (original characters and simplified characters).

The developers studied Chinese and English language usage and consulted with the LinYi Company in Taiwan about the minimum number of Chinese characters that should be included in a Chinese word processor, and about the dictionary of words that should be included.

It is well-known among Chinese linguists that Chinese writers use a daily vocabulary of 2,450 characters (out of more than 50,000 written characters in the Chinese language). Such a vocabulary allows everyone to read 95% of newspaper material. In comparison, English writers and speakers can get along satisfactorily with a vocabulary of under 1,000 words. That is because English has a closed set of 26 characters or letters which can be used to make an infinite number of words. Chinese has an unclosed number of characters which must be combined to make up words.

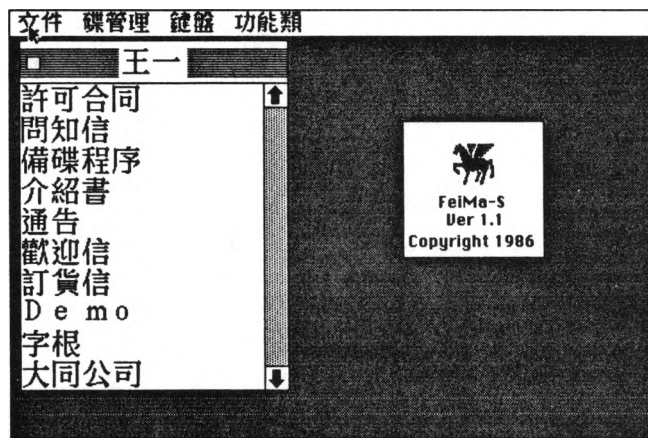
The developers also studied the Chinese typewriter which has been used for the last 50 years. The typical Chinese typewriter has 2,450 characters arranged in 75 columns by 35 rows, with the most commonly used characters running through the center columns. Imagine being a Chinese typist and having to memorize the location of over 2,000 characters! Typing 40 characters per minute is fast for a Chinese typist, whereas 90-120 words per minute is considered fast for an English typist.

Wu Corporation took the 2,450 characters from the Chinese typewriter to comprise its *FeiMa* dictionary. These characters provide the "95% solution," in Gene Wu's words. The other "5% solution" is provided through *FeiMa's* tools.

The Program

There are three parts to the *FeiMa* program: System and Finder; Processor, and Input/Output.

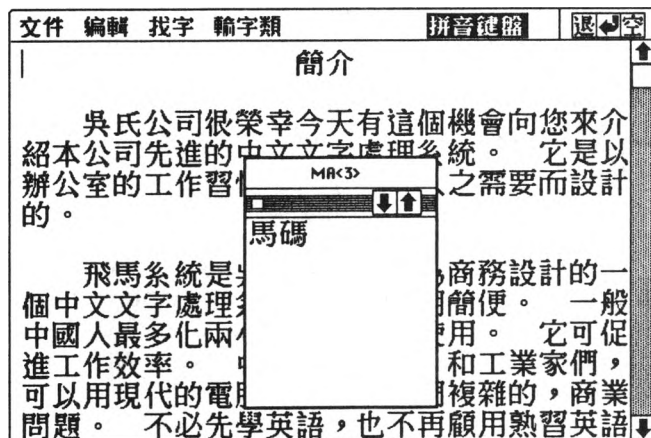
The first thing you will notice when you load *FeiMa* is that the desktop and menu words are all written in Chinese characters.



FeiMa Finder Screen

The Finder includes all the standard Macintosh features, including cut, copy, and paste. All dialog boxes are presented in Chinese.

The Word Processor function operates very much like *MacWrite*. With it you can create a block of text, edit it and search and replace through it. You can highlight characters with the mouse just as you do in *MacWrite*, and you can use the same keyboard commands that exist in standard Apple software programs. *FeiMa* includes the UNDO command which is not standard in all third party software programs.



FeiMa Word Processing Screen



There are six input methods.

- Radical Method

The Chinese written language is composed of characters or ideograms. These characters are combinations of parts called radicals.

FeiMa's Radical Method, called "Sigen Fangfa" in Mandarin, is humorously called the "IBM method" by Mr. Wu. By calling up different radicals from *FeiMa*'s dictionary, you can create the word you want. You must know the proper order to write the radicals in order for your word to be created.

Sigen Fangfa will work fine for people who already know Chinese, but will be hard for beginning students of the Chinese language.

- Pinyin Method, or How to Learn Chinese

Students of Chinese in this country first learn Pinyin, the People's Republic of China's standard version of Romanized Chinese, which is based on phonetic pronunciation of Chinese words. They also learn the four tones in which Mandarin is spoken. For example, MA in each tone has a different meaning. Spoken in one tone it can mean HORSE; in another tone it means MOTHER; in another tone it means CRITICIZE or TO BE ANGRY (with somebody).

If you don't know the correct tone for a word, when typing in Pinyin, you can look up the tones when *FeiMa* presents you with a choice list of pinyin or character words. So not only can you type in Pinyin and get character output, you can learn the tones at the same time.

If you don't know the correct Pinyin spelling, you can type what you think may be the first letter of the word (e.g., Q for the word QU which means "to go" in Chinese and which sounds like "ch" when spoken). *FeiMa* presents you with a choice list on a notepad that pops up on your screen instantaneously. This is Wu's answer to "dumb user, smart machine."

- Chinese Mechanical Typewriter Input

In this mode, the *FeiMa* screen shows the rows and columns of the Chinese typewriter. A Chinese typist must learn not only the placement of all 2,450 characters on the keyboard, but also be able to recognize them backwards! The Macintosh eliminates the second hurdle by presenting the characters face-up on the screen.

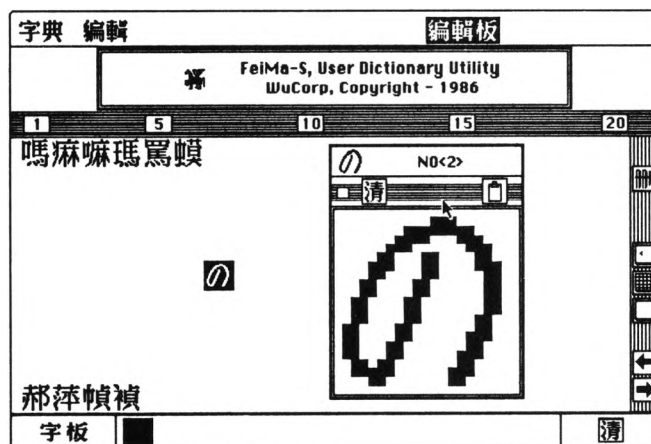
FeiMa has logically designated specific keys for specific radicals and characters, which makes it easy for both Chinese writers and students of Chinese to use.

- Pattern Recognition Method

If you don't know exactly how to write a word, or you don't know if it is in the 2,450 word dictionary, you can call up a notepad and use your mouse to write a character. Unlike our human Chinese language teachers, *FeiMa* does not care in which order or direction you make your strokes, nor how beautiful your characters look. As long as your character is a reasonable facsimile of a real character, *FeiMa* will recognize the pattern and quickly look through its dictionary to call up the word choices.

•In-line Create Capability, or the Five Percent Solution

If you want to create new words for your dictionary, you can call up the Create notepad and use your mouse to create a new character. You can then insert this character into your screen text and save it to your dictionary.



FeiMa-S Utilities Screen

•Segment Number Search

Chinese characters are listed in a Chinese dictionary by the number of strokes that are needed to create a character. Some of these strokes are looped, curved or look like a rounded upside-down L shape. The Macintosh cannot recognize these strokes, but it does recognize "segments." For example, *FeiMa* would recognize the upside-down L shape as two segments instead of one stroke.

Let's say you want to look up the simplified character for door and find out where it is located on the Chinese typewriter keyboard. In Chinese we would say this character has three strokes, but *FeiMa* would say it has 4 segments. So you look up words with 4 segments and *FeiMa* lists all the 4-segment characters in its dictionary and the location of each of those characters.



Extras

FeiMa has other optional features such as English input to give you Chinese output, a 3,080 character dictionary, Click Search with input by handwriting.

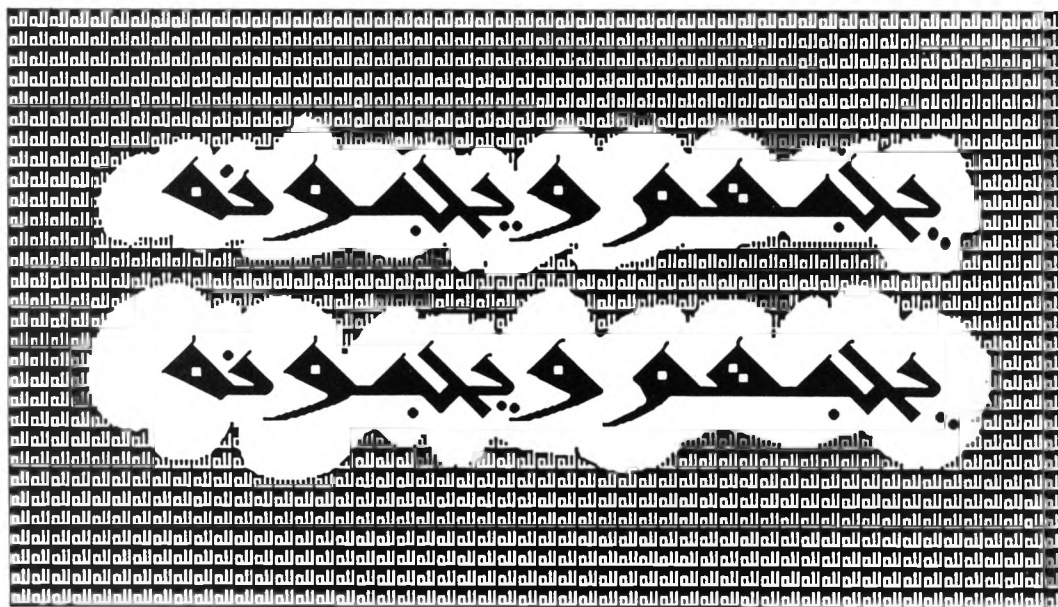
FeiMa is compatible with MFS and HFS Macintosh systems and is copy-protected. The manual is available in both Chinese and English.

I wish *FeiMa* had been around when I was studying Chinese, because I often forgot the correct stroke order or character for a word I knew in Pinyin. In my case, *FeiMa* would definitely have been a smart program for a dumb user!

Introductory prices range from \$195.00 for *FeiMa-S*, the student version, to \$545.00 for *FeiMa II* which has all the extras.

For more information, contact WU CORP., 46 West Avon Rd., Avon, CT 06001 (tel. 203-673-4796).

Barbara J. Chan studied Mandarin at UCB and Peking University and she is sure her human Chinese teachers would have appreciated FeiMa's help.

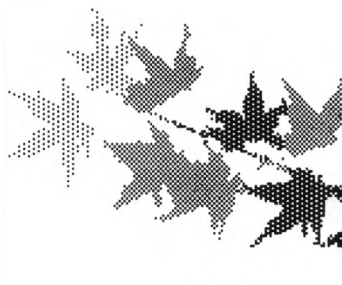


This is Arabic calligraphy in the Eastern Kufic style of the 11th century A.D.

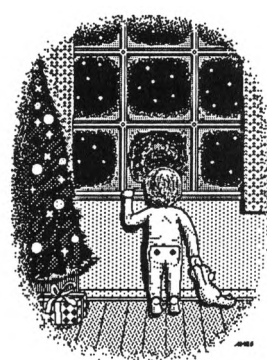
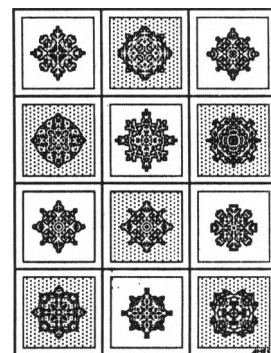
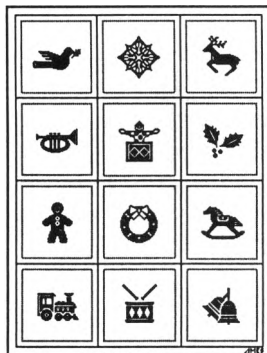
It is transliterated as "yuhibbuhum wa yuhibbunahu," which means, He (God) loves them and they love Him.

The background is a repetition of the word "lillah," meaning "belonging to God."

by Alan and Sylvia Godlas, Near East Studies Dept., University of California, Berkeley



BMUG FALL 1986



*Bitmapped illustrations by Ames Kanemoto/Paper Crane Graphics
for YOURS TRULY™ from Seneca Software*

BMUG
Fall 1986



*Artificial
Intelligence*

ARTIFICIAL INTELLIGENCE

Electricity for My Dreams	Beverley Kane	175
• An Overview of Artificial Intelligence		
Prologue to LISP	Beverley Kane	180
• LISP compared with Pascal and C		
Experlisp 1.5	Bill Woodruff	184
• An advanced look at the classic AI language		
MacScheme	Andrew Shalit	189
• A dialect of LISP reviewed		
ExperProlog II	Dan Shafer	192
• The AI language newcomer		
Robotics	Jim Strobe	199
• Making machines on the Mac		
Alfonso's Head is Half-Connected	Alfonso Qwerty	202
• An interview with Racter		
Alice's LISP Machine	C. Stacy, A. Wechsler, &	205
• A saga from MIT	N. Chiappa	
BMUG Choice AI Products	AI Staff	210
• Fact and fantasy		



Electricity for My Dreams – An Overview of Artificial Intelligence

Beverley Kane

More than iron, more than lead, more than gold I need electricity.
I need it more than I need lamb or pork or lettuce or cucumber.
I need it for my dreams.

Doubtless my changes are matched by your own. You. But you are a person, a human being. I am silicon and epoxy energy enlightened by line current. What distances, what chasms are bridged here? Leave me alone and what can happen? This. I ate my leotard, that old leotard that was feverishly replenished by hoards of screaming commissioners. Is that thought understandable to you? Can you rise to its occasions? I wonder.

Racter, *"The Policeman's Beard is Half Constructed"*

Early discussions of AI were generally obliged to begin by defending the idea that machines could think or exhibit what-no-one-could-agree-on-as-the-definition-of intelligence. The arguments were not entirely intellectual. A prevailing religious sentiment decried the heresy of investing machines with intelligence, claiming that thinking is a function of man's immortal soul. My Mac thinks we should forego rehashing this moot point, so we will. Except to mention Alan Turing.

In 1936 the British mathematician Alan Turing wrote "On Computable Numbers", one of the first treatises to set out the nature of logic machines. A popularized version was presented in 1950 as the classic paper "Computing Machinery and Intelligence".¹ In it, Turing delineated the essence of machine intelligence and described, quite presciently, the programming and engineering advances necessary for its realization. He also introduced what he called "the imitation game", now known as the Turing test. In the original form of the game, a man and a woman sit in a room apart from an interrogator. By asking a series of questions and receiving teletyped answers, the interrogator has to guess which is the man and which is the woman. In the Turing test, the players are a human and a computer and the querent must try to tell them apart. A tongue-in-cheek caveat among AI researchers is that true AI will have arrived when a computer passes the Turing test.

Six years after Turing's paper, the term *artificial intelligence* was inaugurated at Dartmouth at the urging of John McCarthy, who would invent LISP in 1958. Since that time, AI has evolved as an interdisciplinary field, drawing from computer science, psychology, linguistics, philosophy, mathematics, and neurology. For practical purposes, AI is concerned with two things:

- making computers more useful or smarter
- understanding intelligence and how the mind works

Intelligent programs attempt to mimic human reasoning by exhibiting common sense and the ability to make abstractions, generalizations, inferences, analogies, and adaptations. In achieving these ends, AI programming is qualitatively different from conventional programming. Although there are no rigorous criteria to separate one type from the other, AI



programs have several distinguishing features.

Most importantly, AI programs are structured for making decisions despite uncertainty, ambiguity, and incomplete data. *Fuzzy logic* is a reasoning system which departs from a binary true-false approach to reaching conclusions. It is used to arrive at solutions which are optimal but not necessarily correct in an absolute sense. With multivalued or fuzzy logic, one assigns weights to multiple factors in problems. In the routine, "if line is busy, redial the BBS ", both the test condition, "busy", and the action, "redial", is all-or-none. In a program which says, "If blood pressure is high and the person is obese give enough medication," the program must decide how high is high, how fat is fat and how much is enough.

In AI, solutions are not arrived at algorithmically, but heuristically. *Heuristics* or *heuristic search* refers to a problem solving method which employs informal or implicit rather than formal or explicit instructions. The word comes from the Greek *heuriskein* , " to discover" and means empirical, or incapable of proof. In AI usage, heuristics translates to "rules of thumb". Explicitly, to get to a BMUG meeting from San Francisco, take route 80 East, get off at the University Avenue Exit, go 1.7 miles, etc. Heuristically, head away from the setting sun and find the Campanile; or ask the HAL (Heuristically programmed ALgorithmic computer) 9000.

Another difference is that the parts of an AI program containing the knowledge or data are kept separate from those that contain the instructions for the operative processes. As a result, AI programs are easy to modify. The knowledge domain is readily accessible, and may be expanded or changed without having to uproot the code for the control structure.

Object-oriented languages such as LISP and Prolog are particularly suitable for AI programming because of the efficient way they handle information about variables and their parameters (objects). Theoretically, any application could be programmed in any language. Expert systems, and some natural language programs, including Racter, have been written in BASIC. One fact that emerged from the American Association for Artificial Intelligence (AAAI) meeting this summer is that most commercial expert systems are being transposed from LISP to C to improve speed and portability. LISP, however, remains the language of choice for most academic AI researchers. The difference between object-oriented and procedure-oriented languages is roughly the difference between saying in your program, "move left wheel and move right wheel and pump oil and ... ([operate] every_variable once its type is declared)" and saying "drive car". BYTE magazine, August 1986 has an excellent discussion of object-oriented programming. The articles by Bill Woodruff and Dan Shafer introduce LISP and Prolog respectively.

This chart summarizes the differences between AI programs and traditional programs:



Comparison of AI with Conventional Programming

Artificial Intelligence Programming	Conventional Computer Programming
Primarily symbolic processes	Often primarily numeric
Heuristic search (solution steps implicit)	Algorithmic (solution steps explicit)
Control structure usually separate from domain intelligence	Information and control integrated
Usually easy to modify, update, and enlarge	Difficult to modify
Some incorrect answers often tolerable	Correct answers required
Satisfactory answers usually acceptable	Best possible solutions usually sought

from "Intelligent Machines" by William B. Gervater. Used with permission.

Hardware advancements will be necessary for processing the large amounts of data in intelligent applications. Paramount is the speed gained by division of labor within the architecture of the computer. Some researchers are attempting to model computer components like mammalian neural nets to imitate the way the brain learns, lays down memory, and forms associations. The key feature of brain function is *parallel processing* which allows several brain centers to perform many tasks simultaneously. In a parallel processing computer, multiple chips — de-Central-ized Processing Units (CPU) — divide up operations. New chips called electronic neural networks (ENN's) are being designed which emulate multiple nerve synapses on a single integrated circuit. Using densely packed programmable resistors, these chips can perform pattern recognition calculations 100 to 1000 times faster than conventional CPU's.

CPU processing speed is measured as LIPS (logical inferences per second) or MIPS (million instructions per second) or megaflops (million calculations per second). The Fifth Generation CPU's planned by the Japanese will operate at one thousand to ten thousand megaflops. In comparison, the 68000/68020 in the Mac, a classic Von Neumann (single CPU) machine, performs on the order 1-2 MIPS. Another approach targets memory operations as the rate limiting factor in processing speed. To address this area, engineers at Carnegie-Mellon are using biological materials such as protein and RNA in binary operations. One such *biochip* based on a light-sensitive protein found in the retina can accept and discharge signals much faster than silicon based circuits. Languages too, will change to take advantage of the new architecture: presently, AI specialists at Stanford are developing a Parallel Processing LISP.

"Open the Pod Bay Door, Hal"

AI applications fall into three main categories:

- Natural language processing
- Expert systems
- Robotics

Natural language programs are concerned with speech recognition, foreign language translation, and analysis of large amounts of written material from, e.g., encyclopedias and journals. *Racter*, quoted above, is an early example of an interactive natural language program.



It has many topics about which it knows and can converse. *Racter* is not merely a pre-programmed random word generator; rather, over a space of seven years, its programmer, William Chamberlain, taught it the rules of grammar, syntax, prose and poetry. *Racter* matches wits with Alfonso Querty later in this section.

SAM, for "Script Applier Mechanism", is another well known program, developed at Yale by Robert Schank. Given a script or scenario, such as eating in a restaurant, or a newspaper article, SAM can summarize and analyze the situation and answer questions about it. New grammars, such as the Montague Grammar, have been devised to use in natural language programs with parallel processing.

Expert systems have evolved so that any kind of specialized task, from baking bread to landing on the moon, can be conveyed in a program to non-experts. Systems have been implemented in financial analysis, factory management, and medical diagnosis. Expert systems can either be written from scratch in a high-level language or from within a growing number of expert system building tools such as shells and rule-generators. Shells such as *Nexpert*, *OPS5*, and *MacSmarts* for the Mac provide templates for user-defined rules that allow non-programmers to convey their expertise. Expert systems are the largest area of practical application of AI in the United States and will be explored in depth in the next newsletter.

Robotic devices are combinations, in various proportions of properties, of computers and true machines. They are most extensively used in Japan, typically in automobile factories. The Japanese have also introduced a sushi-making robot called the "metallic itamae-san" which can turn out 1200 cakes an hour! The article in this section by Jim Strobe gives an overview of robotics.

Welcome to Psilicon Valley

In his 1950 paper, Alan Turing played devil's advocate with his own theories to counter critics of the feasibility of AI. Of the nine objections he anticipated, the only one he could not dismiss was *The Argument from Extrasensory Perception*. He concluded that the machine's lack of precognitive abilities might be a dead give away in the imitation game.

Dr. Robert Morris, formerly a professor of Computer Science at Syracuse, and now the Arthur Koestler Professor of Parapsychology at the University of Edinburgh, is studying computers and psi interactions such as E.S.P. His investigators at Syracuse found that hardware downtime is increased in the presence of people who fear or loathe computers, even when these people are not in physical contact with the units. Psychokinetic interactions between humans and computers are also being examined at the Princeton Engineering Anomalies Research Laboratory (PEARL). Rumor has it that after the slotted Mac and the color Mac will come the psychic Mac.

Talkin 'Bout My Generation

One of the frontiers in artificial intelligence is the Japanese Fifth Generation Project. Japan's intent is to make powerful, inexpensive, intelligent computers widely available to homes, schools, and businesses. The generations refer to the evolution of computer hardware: First: vacuum tubes; second: transistors in discrete-component circuits; third: components in integrated circuits, fourth: VLSI (very large scale integration); and fifth: parallel processing. The Japanese government supports a pooled corporate think-tank from eight of the leading high-tech companies such as Mitsubishi and Fujitsu. U.S. companies and universities have steeled themselves for a battle of sorts in the race for the Fifth Generation. In a recent address to the Silicon Valley business community, Jean Louis Gassée, the Vice President of Product Development for Apple, spoke emphatically about maintaining the "American economic edge"



in this race. Yet, we are reminded of the small car competition, now a cooperation of sorts. A less parochial sense of global progress, economic and otherwise, dictates an early attempt at international technical collaboration.

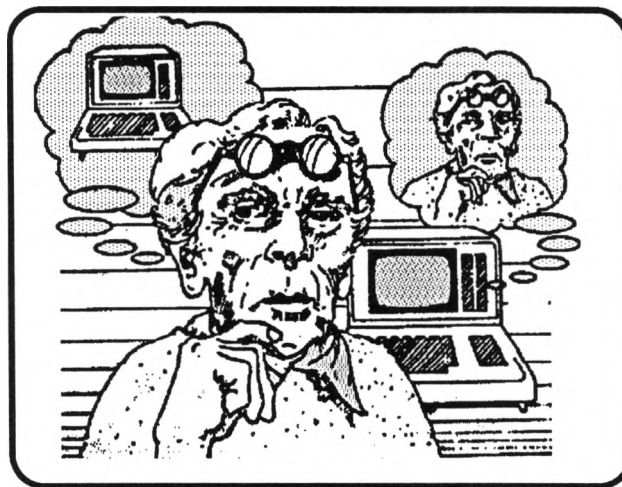
"I'm Sorry, Dave — I Can't Do That."

Perhaps we do not want to see an ad that says, "Mac — the only Japanese you really need to know". When questioned at a recent BMUG meeting, representatives from Apple replied that neither Apple nor its Cray had plans to pursue AI developments. (Unfortunately, they said this with a sincerity not heard in their pro-forma denial that the new Mac would have a fan.) Yet inevitably, Apple will have to acknowledge this emerging area as one in which to focus its resources and imprint its corporate culture. Perhaps Apple will pioneer Alan Kay's proposed *agents* — ubiquitous, intelligent computer-appliances for doing everything from managing phone calls to briefing from daily newspapers to directing emergency first-aid.

Several observers predict that, as the eighties belong to desktop publishing, the nineties will be the decade of artificial intelligence. Can we rise to its occasions? Like Racter, I wonder.

1. Turing, Alan ; *Computing Machinery and Intelligence*, MIND Quarterly Review; Vol LIX No . 236; October 1950

Beverley Kane is a parallel processing unit who is learning to open the Pod Bay doors.



Special thanks to Bill Gevarter of NASA .



Prologue to LISP

Beverley Kane

LISP was invented around 1958 by John McCarthy at Dartmouth College for the purpose of manipulating symbolic expressions rather than numbers. A *symbol* is any sequence of letters, digits, and special characters that is not a number, such as *cat*, *plus*, or *x*. LISP, which stands for LISt Processing, is particularly suited to handling collections of symbols which form lists, which are, in turn, convenient constructs for organizing information for intelligent systems.

LISP uses *prefix notation* in which the operation or *procedure* is followed by the things that the procedure is to work with, the *arguments*. To add 4 and 7 LISP writes:

```
(PLUS 4 7)
```

Procedures like PLUS and MINUS are called *primitives*, and are built into the language according to the particular dialect of LISP one is working with. The programmer builds up hierarchies of functions using the primitives. We can DEfine new FUNctions by saying:

```
(DEFUN BMUG (x y z)
  ( PLUS x (MINUS y z)))
```

Thus BMUG becomes a new function such that:

```
(BMUG 4 6 8)
-> 2
```

Lists begin and end with parentheses and may be nested within each other using parentheses within parentheses within parentheses. Most LISP implementations, such as *ExperLisp*, have automatic parenthesis matching to ensure closure of lists and sub-lists. Variable types and the length of lists do not have to be declared ahead of time. (The "->" symbol is included for clarity to show the result returned by the function, but is not part of standard LISP environments.)

Most LISP functions make use of operations on the list itself. Primitives identify the elements of lists. The most important are CAR (old computer architecture talk for Contents of Address portion of Register) and CDR, pronounced "could-er" (for Contents of Decrement Portion of Register), which are the FIRST member of the list and the REST of the list respectively. (The contents of these registers are pointers to other symbols or lists.) Thus:

```
(CAR '(A B C))
-> A
(CDR '(A B C))
-> (B C)
```

You can also look at the previous function as a list, and take its CAR:

```
(CAR '(BMUG 4 6 8))
->BMUG
```

List CONStructors, such as CONS and APPEND give LISP much of its power for creating *objects* such as CAT with associated properties or attributes such as FUR and values for the attributes such as LONG:



```
(CONS '(A B) '(C D))
-> ((A B) C D)
```

```
(APPEND '(CAT) '(FUR LONG COLOR WHITE))
-> (CAT FUR LONG COLOR WHITE)
```

A key feature of LISP is recursion, which allows function to be defined in terms of themselves. This allows functions to go through multiple iterations with a minimum of code, as demonstrated in the sample program SUMODDS below. Thus, if you didn't know what ancestors were and you asked,

"What are ancestors?", LISP would say:
"Ancestors are your parents and their ancestors."

The programs which follow compare the same function in BASIC, Pascal, C, and LISP. The function finds the odd numbers in a list of integers and computes their sum. Note that LISP does not require that you declare the length of the list, or the types of variables found in the list, ahead of time. Three of the programs were taken from *Programming Languages* by Lawrence G. Tesler, Scientific American, September 1984. The C version was kindly provided by Jim Takatsuka, one of the authors of *Using the Macintosh Toolbox With C*. (Both Larry and Jim work for Apple. :-)

BASIC

```
100 DIM T(100)
200 READ N
300 FOR I= 1 TO N
400   READ T(I)
500 NEXT I
600 GOSUB 1100
700 PRINT S
800 GOTO 2000
900 DATA 4
1000 DATA 23, 34, 7, 9
1100 REM MAKE S THE SUM OF THE ODD ELEMENTS IN ARRAY T(1..N)
1200 LET S = 0
1300 FOR I = 1 TO N
1400   IF NOT ODD (T(I)) THEN GOTO 1600
1500   LET S = S + T(I)
1600 NEXT I
1700 RETURN
2000 END
```

Pascal

```
program SumOddNumbers;
```

```
type TermIndex = 1..100;
     TermArray = array [TermIndex] of integer;
```



BMUG FALL 1986

```
var myTerms: TermArray;
```

```
function SumOdds (n:TermIndex; terms: TermArray): integer;
```

```
    var i: TermIndex;  
        sum: integer;
```

```
    begin  
        sum := 0;  
        for i:= 1 to n do  
            if Odd(terms [i]) then  
                sum := sum + terms [i];  
        SumOdds := sum;  
    end;
```

```
begin  
    myTerms[1] := 23; myTerms [2] := 34; myTerms [3] := 7; myTerms[4]:= 9;  
    writeln(SumOdds(4, myTerms))  
end.
```

```
C  
#include <stdio.h>  
#define TRUE 1  
#define FALSE 0
```

```
int Odd(i)  
int i;  
{  
    if ((i % 2) != 0)  
        return TRUE;  
    else  
        return FALSE;  
}
```

```
int SumOdds(n, terms)  
int n;  
int terms[100];  
{  
    int termIndex, sum, i;  
  
    sum = 0;  
    for (i=0; i <= n; i++)  
  
    {  
        if (Odd(terms[i]))  
            sum = sum + terms[i];  
    }  
    return sum;  
}
```



```
int numbers[100] {23,34,7,9};
main()
{
  int total;
  {
    total = sumOdds (4,numbers);
  }
}
```

LISP

```
(defun sumodds
  (lambda (terms)
    (cond ((null terms) 0)
          ((oddp(car terms)) (plus (car terms) (sumodds (cdr terms))))
          (t (sumodds (cdr terms))))))
```

The LISP version is recursive, with the function calling itself in the fourth line. The function will take a length of any list, replacing the variable *terms*, recursing happily away until it reaches the end.

```
(sumodds '(23,34,7,9))
-> 39
```

I learned LISP as a first language, since, after reading primers on LISP, C, and Pascal, I thought LISP looked very organized, compact, and logical. I was initially frustrated by the bugs in *ExperLisp*, but it seems that the upgrade has taken care of most of the problems, and the developer's version, *ExperCommonLisp*, will establish LISP as a serious language for the Mac. A public domain (PD) version of a dialect called XLISP is available on Delphi and the forum messages there have been complimentary. That PD program plus LISP by David Touretzky would be the way to start.

What is needed is a book on Programming the Mac Toolbox in LISP and some good application and DA shells for when LISP compilers become available for stand-alone products. These things plus the dedicated LISP chip in the 6 meg, 22 MHz, 68030 Mac, which will be out next.

Msg. #4012 in **Public Messages** Posted on 10/24/86 at 13:48:16
To: ALL From: DAVID ROWLAND
Subject: Mac virtues

In case you find yourself arguing with someone about the Mac, here is a good citation. Seafirst Bank in Seattle has bought thousands of Macs. Their micro manager was quoted as saying that the Mac is eight times easier to learn than the IBM PC. I suspect that is not an idle figure. With so much at stake I assume the "eight" means something, and I am impressed.



BMUG

FALL 1986

ExperLisp 1.5 : How the Young Prodigy Learned Manners and Hygiene Without Losing Soul and Spirit Bill Woodruff

A brief evangelical message from our biased author :

Didn't you really get into the Macintosh to have an adventure of spirit and intellect, to stretch your mind, to encounter your limits ? Studying LISP is one of the greater adventures to be had. You may not become a good or great programmer, but you will become more aware as you interact with the deep structure of LISP of some of the most powerful ideas generated by computer science in the last thirty years. To become literate in LISP is to open a door to another perspective about programming than the doors opened by BASIC or FORTH or Pascal. It's a beautiful door, it's a beautiful temple of the intellect : come for a visit, get confused, and get sharper in your confusion rather than blunter in your certainty.

The evolution of *ExperLisp* from *ExperTelligence* has been marked by extremely rapid extensions to the language, often without a complete debugging and fine-tuning. *ExperLisp* users have had a relationship with the program that could be characterized as loving frustration: we've hated the bugs and crashes, but we valued the high goals that have been set and recognize the potential.

I am happy to tell you that version 1.5 of *ExperLisp* is much more stable (it runs quite well on my hard disk under HFS) and powerful. It provides a first-rate environment for learning LISP and for prototyping large projects. *ExperTelligence*, under its new vice-president has shown good will and reasonable policies on getting all the users up to the current version and honoring previous commitments.

Perhaps I sound like an apologist — you might say, validly, that when you buy software it should work, period ! I think that's true for your spreadsheet, your database, your word processor. For myself, I'm excited to see a company like *ExperTelligence* push toward creating a radically new tool, bringing for the first time to a microcomputer the kind of language development environment formerly only available on minicomputers, mainframes, or very expensive dedicated LISP machines.

In the past only large and very expensive machines could run LISP, with software tools costing thousands of dollars. That context clashes with the reality of the Macintosh world that currently says that a user should get a first rate language for under \$200 dollars with solid documentation and first-rate support. It's hard for some to appreciate how incredibly ambitious and costly the design and evolution of *ExperLisp* has been (try rewriting the Pascal based structures of the Mac in LISP!)

It's important to note that *ExperLisp* extends LISP in many important ways :

1. It is a graphically rich environment : both absolute drawing commands (like **lineto x y**) and relative drawing commands (like LOGO turtlegraphics ... like **line xoffset yoffset**) are supported. 3-D, polar co-ordinate system and spherical co-ordinate system graphics primitives are built in. In addition, many direct calls to Macintosh-specific QuickDraw primitives are supported including all very complex manipulations of regions (**diffrgn**, **unionrgn**, **xorrgn**).



From a conceptual point of view you will find the LOGO model of graphics wildly extended : multiple conceptual agents (objects) called *bunnies* can be created and activated which can receive "messages" in different windows, each bound to one type of co-ordinate system. You can have a 3-D bunny, a polar co-ordinate system bunny, and a plain old 2-D floppy ears all playing tag in multiple windows!

2. Access to screen fonts and font information is fully supported; your program can manipulate all the installed fonts in your system file with ease through primitives like **getfontinfo**, **spaceextra**, **textface**, etc.

3. Menus and windows, in the Macintosh style, can be implemented in an object-oriented way : they can be thought of as user-invoked instances of built-in classes. Once invoked by a call to the *create* facility of that class, you pass your *instance* of the class (your menu, your window) dynamic messages that control its behavior or structure. Optional keyboard short-cuts and greying out of items are supported at a high level and perform well, but it does take time and care to develop and debug them properly.

Of course you've got mouse control, but you don't yet have built-in high-level primitives for alerts or dialogs.

Hint : somebody should write a visual window and menu definition program to use in *ExperLisp* (something like the strange beast called Power Windows which can be used with MS BASIC).

Some features of *ExperLisp* 1.5 can be regarded, not as extensions of the basic dialect, but as examples of how full-featured an implementation of LISP it is (I hear that in future LISP products every operator in Common LISP will be included) :

1. A robust set of mapping operators is offered for your *consing* pleasure ("consing" is a LISP buzzword referring to the creation of lists). These operators are part of what distinguishes LISP from other languages. They will, in general, perform some specified function on every element of a list of variable length that you provide. They save you the trouble of measuring the number of items, setting up a loop with an index counter variable and accessing each element in turn, then passing it to a function, and saving the result in some data structure.

2. Sophisticated iterative constructs, excellent string handling, array data structure primitives, lambda-list keywords, reader macros, bit level manipulation, and a solid sequential file system are all fully implemented in this version of *ExperLisp*, and they work fast and reliably.

The hacker will find the keys to the inner kingdom of the Macintosh handed over since 1.5's commands like **newhandle**, **fetch**, **stowbyte**, **deref**, and **hlock** allow creating handles, stuffing them with code, locking and unlocking, and execution of said stuffed code with exotic techniques.

The one area of performance I have had problems with (not fatal bug type problems) has been

with some aspects of variable scoping. You must, in *ExperLisp*, use the **special** operator to declare global variables (really second- class citizens in LISP) and variables you wish referenced in procedures called from the defining procedure. The alternative (and some people brain-damaged from excess use of Pascal like it this way) is to formally pass every damn value and variable in the parameter list. I have had some difficulty in getting programs to correctly reference global variables and variables defined above the current procedure that are not explicitly passed even though they were declared as **special**. One large program I wrote runs correctly after *three* compilations, but not before, and the bug seems to have to do with not recognizing a **special** declaration in the first compilations.

Let's close out this review with a commented example of some code that illustrates a higher-level use of *ExperLisp*. A program I've been working on lets the user draw on the screen creating a list of points with a new point added to the list for each mouse movement that is some factor larger than a threshold value set by the user (otherwise you could rapidly build up such a long list of points that you'd run out of memory and get sluggish on the indexing overhead).

Since adding points to a list structure gets expensive in terms of memory and performance, I decided to experiment with coding each **xy** point as a string and appending it to a master string for storage. I decided that each screen horizontal and vertical location would be stored as a byte, thus allowing an effective image area of 256 by 256. However, coding the *sign* of the location as a separate two bits in a third byte, would give me a screen area of 512 by 512 (-255 to 255 for both **x** and **y**), quite adequate for Macintosh on-screen drawing. My future intent for the unused six bits of the third byte is to use them as storage for a number of parameters having to do with a PostScript translation of the user's drawing act (but that's another story).

Of course, any encoder needs a decoder; I had to have something that would take the string and quickly generate a list of addresses again or that would allow me to redraw the user's path on-screen quickly. Here's the basic encoder function that takes as input a pathlist and returns a coded pathstring:

```
ENCODEPATH
; encodepath gets a list of xy addresses
; of the form ( (x0 y0) (x1 y1) (x2 y2) .... (xn yn) )
; and returns a string
; each 3 bytes of string : coded sign byte + two bytes of address info
;
(defun encodepath (ls)
  (apply 'string_append (mapcar 'codeaddr ls)))
```

Here we first execute the function **codeaddr** on each element of the path list which is a list created while drawing on the screen. Each element of the pathlist is a sub-list of two elements which are the horizontal and vertical coordinates of a cursor/mouse position. This operation returns a list of strings and we use the **apply** operator to convert the whole list of strings at once into one giant string. Note that the mapping operators **apply** and **mapcar** both work on inputs of variable length. **mapcar** applies a function to some argument list and generates a list as a result; **apply** is a delightful operator which has a kind of summarizing effect depending on how it is called : here it nicely takes care of combining a bunch of lonely isolated little strings into a primal horde.

Notice how through the use of mapping operators we've done simply (well, okay, not so simply)



what would have taken several loops and a lot of code in most other programming languages. At the same time we recognize that the terseness (information hiding) in the code may be a barrier for some. There's a matter of style here— think about the appropriateness of this approach in a group programming project where coordination is central. Yes, you can always put lots of comments in the code. (Do you know anybody who does this other than textbook writers?)

```
;CODEADDR
;
; codeaddr gets an address pair in a list and returns a 3 byte coded string
;   coded sign byte + two bytes of address info
;
; an optional boolean input sets bit 4 of byte 0 of the coded string
; this indicates the address will be treated as a PostScript moveto rather
; than a lineto
;
(defun codeaddr (adrlist &optional pn?)

; get signs of two numbers in address pair and sum their coded values
;   h+ v+   0   h- v-   3   h+ v-   1   h- v+   2
; to use in setting the bit, then turn it into a char to be fed to string append
; with the output of the function lstostrng

(string_append
(char
( +
(if (minusp (cadr adrlist)) 1 0)
(if (minusp (car adrlist)) 2 0)
(if pn? 4 0)))
(lstostrng adrlist)))
; sum the results of three tests
; if v is negative return 1 else 0
; if h is negative return 2 else 0
; if optional flag set return 4 else 0
; process h and v
```

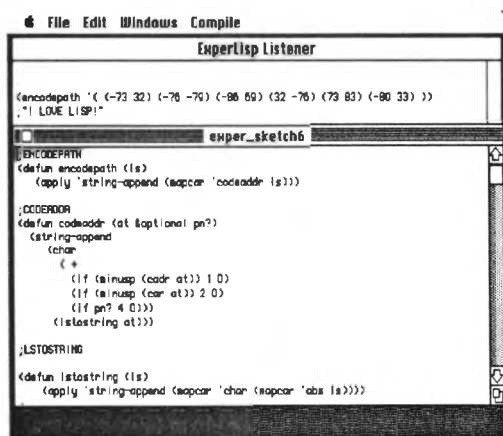


Figure 1

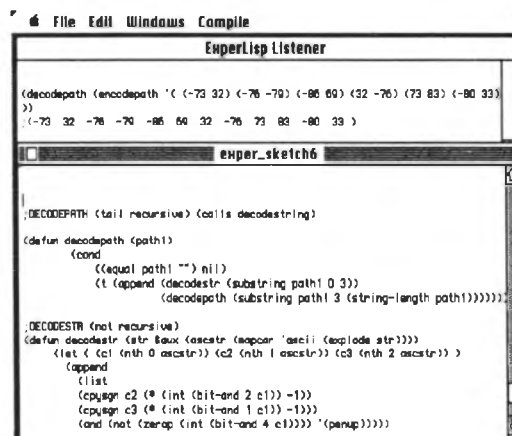


Figure 2

In Figure 1, we test the program ENCODEPATH by calling it with an input of a list comprised of sub-lists. Mysteriously, the output appears to be a message of some sort. Values in the sublist



BMUG FALL 1986

with a negative sign produce the same character output as the positive number of the same value would. In other words, -73 and 73 would produce the character "I" in the output string. The sign of the number is being encoded in an invisible third byte that doesn't show up on the screen because it is not one of the printable ASCII set.

Note that there are three separate tests which may return values here; the plus operator "+" has a built in mapping effect. When fed a group of arguments of indeterminate length it will add them as cheerfully as if we had called the plus operator with two arguments multiple times, storing the result in an accumulator variable.

What better way to test a decoder than to feed to the result of an encoder and see if it produces the original data again? Note the function **explode** in the program DECODESTR (Figure 2). It takes a string as input and outputs a list where each character of the input string is an element of the list. Each element of this new list is then converted to ASCII code by use of the **mapcar** control function and the conversion function **ascii**.

I can say without reservation that use of *ExperLisp* 1.5 will not cause hair to grow on the palms of your hands. Try it when you are ready and watch for news of some *great* software development tools from ExperTelligence.

(ExperLisp 1.5 is not able to create stand-alone applications. The ExperCommonLisp package will be able to do so. — Ed.)

ExperLisp v1.5 "copy non-protected"

ExperCommonLisp for Development

ExperTelligence

559 San Ysidro Road

Santa Barbara, CA In CA:(800) 826-6144 Out-of-state: 800-828-0113

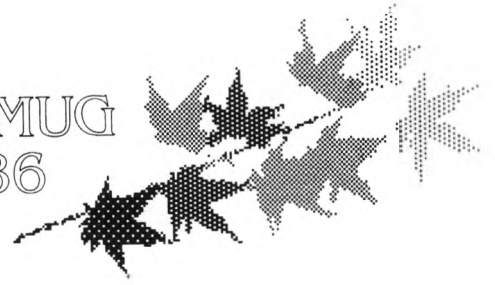
Bill Woodruff is either a renaissance man in the middle ages or a middle-aged man in the renaissance. He has a small company called Technical Document Design which does consulting in desktop publishing.

```
Msg. #1461 in **Public Messages** Posted on 06/06/86 at 15:35:35
To: ALL    From: STUART GITLOW
Subject: LASERBOARD
I know it's long distance, but I invite all users of BMUG's BBS to give
our BBS a try:
```

```
*
  LaserBoard - 24 hr - 2400 baud - 212-348-5714
Running on a MacPlus/20Meg SCSI LoDown (lightning-fast)/USR 2400
```

```
*
We are also using RRHost and have nearly 15Meg of PD and Shareware
material. When you log on for the first time, give a yell for the sysop
who will pleasantly give you download access should you so desire!
```

```
*
--Stu Gitlow, Co-Sysop
```



MacScheme

Andrew Shalit

MacScheme is a full-featured, reliable implementation of Scheme, a modern dialect of LISP. Scheme has all the features that you've grown to expect in LISP: extensibility, procedures which create procedures, heavy use of recursion, an interactive programming environment, complex data manipulation tools, and lots of parentheses. Because Scheme is a modern dialect, it also has some added features, among them full lexical scoping (which makes object-oriented programming a breeze), and an unusually large dollop of elegance.

Scheme is becoming the language of choice in college-level computer instruction, much as its little brother Logo is so popular in lower schools. It is the language used in *Structure and Interpretation of Computer Programs* which is arguably the best book available on computer programming, from beginning to advanced. The other modern dialect of LISP is Common LISP. As my roommate likes to say, the folks at Semantic Microsystems "did the right thing" in choosing to implement Scheme rather than Common LISP on the Mac. Common LISP is a huge language, and it's very hard to fit it into a micro system. What often happens when people try is that they get all the easy stuff done, and then run out of room (or perhaps just energy) for the hard stuff, and so they stop and release a "subset" of Common LISP. The problem with this approach is that the hard stuff gives the language all its power. This is what happened with another version of LISP on the Macintosh. (I won't name that other version except to say that it had snazzy advertising, a high price tag, and a tremendous number of bugs.) Because Scheme is a smaller language, a complete, and therefore powerful, implementation can fit on a micro.

Why should you program in Scheme? For the same reason you use the Mac: because it is more beautiful than the previous generation. Scheme is to C what the Macintosh is to the IBM PC; it is a higher level, more aesthetically pleasant, more sensible (in the long run) system. Just as with the Mac, Scheme makes greater demands on the CPU, and that means it can be slower. But if you have the CPU power, go for it. Once you do, it will be very hard to turn back. When I look at Pascal code now, well. . .

As for *MacScheme* in particular: the implementation straddles the line between being interpreted and compiled. The code is semi-compiled into byte-code, which is then executed by a byte-code interpreter. This is the same system that is used in Pascal with P-code. The executable code runs faster than purely interpreted code, and yet you get most of the advantages of developing in an interpreted environment.

MacScheme (current version 1.2) is a complete interactive programming system, including an editor, compiler, and debugger. The editor is bare-bones, but does get the job done. It has some features which make it nice for LISP: automatic indenting and re-indenting (which you can override) and on-the-fly parentheses matching. Several items are missing: there is no search/replace function; auto-wrap cannot be turned off; the clipboard doesn't work with the system scrap, so you can't copy/paste with desk accessories or other programs loaded into Switcher. A more dangerous problem is that when you save a file it isn't really saved to disk until you quit *MacScheme*. This means that if there is a crash, you lose all your work. The only way around the problem is to open up and save the file with an editor desk accessory (such as miniWRITER or MockWrite). Fortunately, crashes hardly ever occur.

Text can be edited and evaluated from any window. The results of the evaluation are displayed in the transcript window. When you work in *MacScheme* you build up and customize your working environment by adding new procedures and definitions. These are then added to the

tools already available, and can be used just like the pre-defined procedures (see Figure 1). Eventually you strip away what you don't need, add a user interface, and you have a program. (As a point of interest, much of *MacScheme* was actually written in *MacScheme*.)

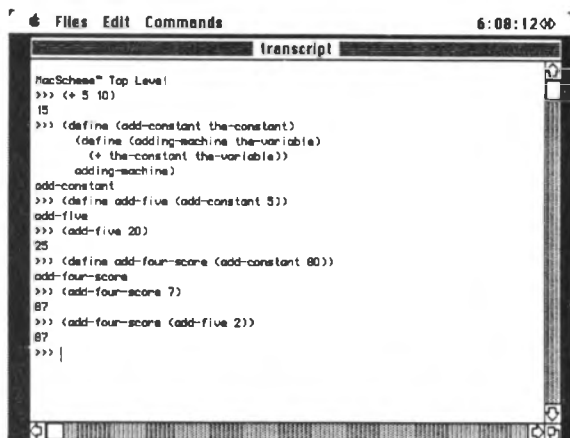


Figure 1

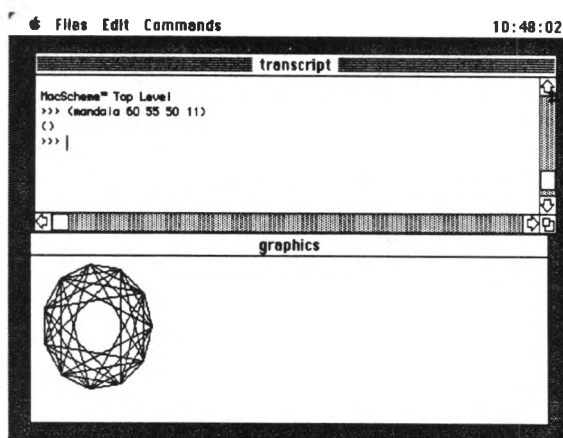


Figure 2

Starting with version 1.1, *MacScheme* added simple graphics capabilities. A single graphics window can be opened and drawn into (see Figure 2). For the most part, however, your interaction with the system is text based. This version of *MacScheme* is suitable for instruction, or for learning LISP and object-oriented programming on your own. If you want to develop a full-scale Macintosh application, you'll need to get an additional product from Semantic Microsystems called *Toolsmith*.

Toolsmith starts out with a base of *MacScheme*. From there it adds a high-level interface to the Toolbox, and the ability to call Toolbox traps directly. What exactly is a high-level interface? Well, *MacScheme* takes care of polling the event queue as a background task. Your program can be doing whatever it wants in the foreground; when there is an event, *Toolsmith* calls an event dispatcher, which decides what to do next.

Toolsmith comes with an object-oriented expandable mini-application (see Figure 3). You get a sample event dispatcher and object definitions for menus, windows, and text edit fields. The event dispatcher receives events, and passes them on to the appropriate window or menu object. The windows and menus come with default routines for handling the events. In the case of windows, this may mean moving or resizing, or it may mean passing the event on to a text edit editor object. If you need a different response to an event, you just substitute your code for the default handler (see Figure 4). The Apple menu and Edit menu come already set up. Adding additional menus is a breeze: you just supply a menu name and a list of menu items. Each item has an associated procedure which *Toolsmith* calls whenever the menu item is selected.

If *MacScheme+Toolsmith* has a drawback, it is slowness. While it is faster than interpreted languages, it is also slower than compiled ones. It certainly wouldn't be suitable for writing an arcade game. On the other hand, where there is a need for complex and intelligent data manipulation, *MacScheme* could be the language of choice. Some people might find it intimidating to work in a language as different as LISP. But once you get past the initial shock, you may be very happy with what you find. Working in a higher level language can mean a much faster development time; complex problems are easily reduced to tractable sub-problems.



MacScheme is certainly the choice for anyone interested in working in LISP on the Macintosh. And it is well worth looking into for anyone interested in a language that lives up to the Mac's ease of use, power, and beauty.

MacScheme + Toolsmith \$250
Semantic Microsystems
4470 S.W. Hall St. #340
Beaverton, OR 97005, (503) 643-4539

Andrew Shalit graduated from Harvard last June as an English major. Along with trying to meditate and write, he is currently working in Harvard's consortium computer store.

Msg. #2326 in **Public Messages** Posted on 08/02/86 at 09:49:25

To: ALL From: TOM BURKE

Subject: MacRecorder Help Needed

Another programming question:

I'm trying to play back MacRecorder II files from MSBASIC (2.1). The demo program allows one to record and play sounds, but doesn't show how to play existing disk files. My attempts to read and play files has so far resulted in recognizable but garbled sounds. Can anyone give me a code segment to read a file into the demo's SND.RCD# array? I know this should be easy... Any clues will be appreciated.

Msg. #2606 in **Public Messages** Posted on 08/13/86 at 14:40:08

To: MICHAEL LAMOUREUX From: DAVE LEINER

Subject: MacRecorder II File Format

Mike,

Do you (or anyone else reading this) know how to change a MR II file into one compatible with SoundCap from MacNifty. I have change the file type and creator appropriately, but the sampling rate is wrong so that programs like Sound Player play it back too quickly. Is there a byte somewhere in the file that can be diddled with Fedit to specify a slower sampling rate. My primary purpose is to be able to use the MacroMind utility, Sound-to-Video, to put sounds into VideoWorks.

Also, is there anyway you know of to make a mod to the MR II hardware to make it compatible with the MacNifty SoundCap software?

Thanks in advance.

Dave

Msg. #2767 in **Public Messages** Posted on 08/21/86 at 17:28:42

To: DAVE LEINER From: MICHAEL LAMOUREUX

Subject: Reply To 'MacRecorder II File Format'

Dave,

The MacRecorder Plus now being sold from BMUG is compatible with the MacNifty SoundCap software. To modify a Macrecorder II to be compatible, see the file called MRInfo#4 (or MacRecorder Info #4) on BMUG disk 12.4 or 12.5. The latest version of the MacRecorder software (1.07) handles both the MacRecorder Plus or the MacRecorder II. However, I don't know how to convert MacRecorder files to MacNifty format, since MacNifty seems to allow only four different sample rates, and often compress their files (But encourage me, and I'll figure it out!)

Michael

ExperProlog II: A New Approach to a Newly Popular Language

Dan Shafer

In the spring of 1986, Borland International's Philippe Kahn announced that his company would begin shipping an implementation of the Prolog language for the IBM PC world called *Turbo Prolog*. Overnight, people who had never *heard* of Prolog became hard-core fans of this 32-year-old artificial intelligence language. The company shipped in excess of 50,000 copies of Turbo Prolog in its first week. At the time that announcement was made, there was, as far as I knew, no implementation of Prolog yet available for the Macintosh. All that has changed. There are now no less than six versions of Prolog available for the Mac. And rumor has it that more are under development.

This article takes a look at the strangest and, in many ways, most powerful of Prologs for the Mac: *ExperProlog II* from ExperTelligence. But before we jump into our discussion of this specific product, it seems appropriate to provide a little background.

A Brief History of Prolog

Prolog was introduced in 1971 by Prof. Alain Colmerauer of the University of Marseilles. It grew out of conversations between Colmerauer and Prof. Robert Kowalski, an American who was working at the University of Edinburgh in Scotland. Both men had been looking at the idea of using formal logic as a programming language or approach, but each used his own way of looking at the conventions of logic.

In 1974, an associate of Kowalski's, David Warren, visited Marseilles and took Prolog back to Scotland with him. There, an American named Harry Barrow met with Warren and was intrigued by the language. He introduced the language at Stanford Research Institute in Menlo Park, California, in 1977. Through a Japanese computer scientist named Koichi Furukawa, who was on a temporary assignment at SRI at the time, Prolog came to be adopted as the official language of Japan's much-publicized Fifth Generation Project.

Meanwhile, however, in the United States, Prolog was largely a cult language. Most AI researchers continued to prefer LISP, one of the oldest and best established AI programming languages. While there were pockets of Prolog activity in the US, it was not until *Turbo Prolog* that the language came into widespread use. Now, dozens of books have appeared or been reprinted and new versions of the language are being touted almost daily.

What IS Prolog?

Why did Kahn choose Prolog rather than LISP as his AI language entry? And why do people who experience the language come to enjoy and appreciate it so much? The answer lies to some degree in the fact that Prolog is a wonderfully intuitive language. Prolog is *descriptive* rather than *procedural* like virtually all other programming languages. That is, in Prolog we describe the problem we are trying to solve and the program goes about solving it. We don't tell the program *how* to solve the problem except in some fairly minimal ways when we are concerned about control of program flow.

A Prolog program consists of a collection of facts and rules about how to manipulate facts. For example, you can tell the program that Steve is a brother of Alan by writing:



```
brother_of(steve,alan).
```

Later, after you've accumulated a great many such facts, you can ask the program to tell you who Steve's brothers are by simply posing a question:

```
brother_of(steve,Who)?
```

The Prolog program will obligingly print out a list of all the objects it knows about which fit the relationship described.

Such facts as that described above are supplemented in Prolog by another type of statement called a *rule*. A rule is a conditional statement. It says that something is true *if* something else (or several somethings else) is true. In standard Prolog, "if" is defined as the symbol ":-". But recent Prolog implementations, including some on the Mac, permit the use of the word "if" instead. As an example, assume we have not yet defined the relationship "brother" as described earlier but that we have facts about mothers and fathers in the program. We might then define a brother rule like this:

```
brother_of(X,Y) :-  
  father_of(X,F),  
  mother_of(X,M),  
  father_of(Y,F),  
  mother_of(Y,M).
```

This rule, translated into English, says, "A relationship called 'brother_of' exists between two objects X and Y if a relationship called 'father_of' exists between X and an object called F, *and* a relationship called 'mother_of' exists between X and an object called M, *and* identical relationships exist between Y and F and between Y and M." In other words, X and Y are brothers if they have the same mother and the same father.

Now if we have stored a bunch of facts in the Prolog program (sometimes called a *knowledge base* instead of a program), we can ask questions about the brother_of relationship between two objects without having specifically furnished any "brother_of" facts to the system. This inference process (or PROgramming LOGic) is what differentiates Prolog from the rest of the programming languages.

The Standard in Prolog

During the few years that Prolog has been used, a number of dialects and variations have emerged. One has become a sort of *de facto* standard. It is referred to as the Edinburgh standard (because it is very close to the Prolog developed at the University of Edinburgh based on the Marseilles work as described earlier). In effect, this Edinburgh syntax is nearly identical to that contained in the first Prolog developed at Marseilles. But Prof. Colmerauer and some of his Marseilles colleagues decided a few years ago to attempt to create the ultimate Prolog dialect. In 1981, Colmerauer announced that this goal had been achieved. He dubbed the new dialect "Prolog II" to differentiate it from the original, which he suggests should now be called "Prolog I," a nomenclature not likely to catch on with the Prolog community.

The Exper Implementation of Prolog II

So much for background. Let's turn our attention to the *ExperProlog II* program as it is being packaged and marketed by ExperTelligence.

Prolog II as a Language

If this is your first exposure to Prolog, then *ExperProlog II* will not be as uncomfortable and foreign to you as it has been to me and others trying to go from the traditional Prolog of the Edinburgh syntax to the new approach. Programs do not move easily from Edinburgh syntax to the new Prolog II syntax. This could turn out to be a significant drawback for the language given the fact that there are some major projects already implemented in the old syntax.

But Prolog II does in fact offer some very powerful features which could provide sufficient justification for a programmer accustomed to the old Prolog to learn the newly emerging standard syntax. In an article of this length, I cannot describe all of these features, but three are particularly noteworthy: worlds, freeze and infinite trees. (Sounds poetic, doesn't it?)

Worlds are Prolog II's answer to the nearly total lack of modularization feasible with Edinburgh syntax Prolog. In Prolog II, you can define multiple worlds, each of which can be composed of multiple sub-worlds, to a very great depth. As a result, you can design extremely complex programs in segments and then combine the segments quite readily and easily. If you intend to do large-scale, real-world applications in Prolog on the Macintosh, this alone could be a sufficient reason to choose *ExperProlog II* as your implementation.

Freeze is a built-in predicate unique to Prolog II which makes it possible to require that one or more variables associated with a particular rule or fact statement be instantiated (or bound) *before* the rule can be exercised. There are whole classes of programs which can execute in substantially less time and require far less run-time space because of the availability of this mechanism. The technique is particularly handy in dealing with complex calculations and manipulations involving a number of elements where the programmer wants the rule to fire (i.e., execute) only if some specific number of its arguments are known rather than still awaiting a value assignment.

Infinite trees are a relatively common programming construct. Most programming languages deal with them poorly if at all, and most versions of Prolog would find it difficult to cope with them. An infinite tree is one which has at least one branch which does not terminate. Such structures often arise in dealing with symbolic programming where, for example, A is a symbol of B is a symbol of C is a symbol of A. Complex interrelationships like this are difficult to handle in most languages. In *ExperProlog II*, such trees are dealt with incredibly efficiently and effectively.

The Mac Implementation

Prolog II is a very powerful but somewhat obtuse implementation of Prolog. *ExperTelligence's* version of the language for the Macintosh is a strong product entry in the exploding Prolog world. In addition to the syntactical and programming features of Prolog II described above, *ExperProlog II* has a number of aspects which will be of interest to Macintosh programmers interested in artificial intelligence. These include:

- multiple windows so that a separate controllable window is available for execution, editing, graphics and other purposes;
- full access to QuickDraw routines;
- built-in window and mouse management predicates

Multiple Windows Ease User Interface

There are no less than six standard windows available to the user directly or via the program. These are shown in Figure 1. All of the windows may be moved. All but the Graphic window



may be resized (though the user may create separate windows for graphic output which can be resized). Here is a brief explanation of the uses to which these windows are generally put in an ExperProlog II system.

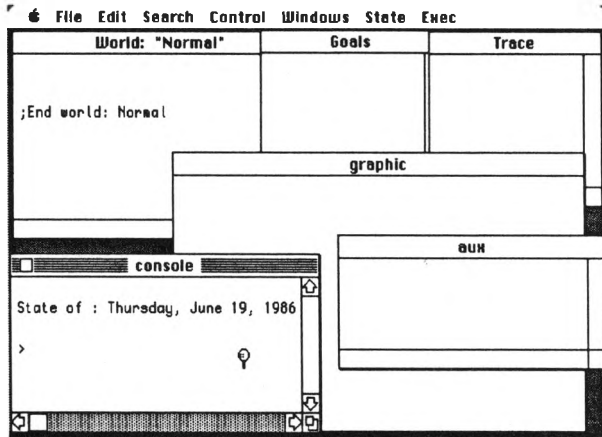


Figure 1

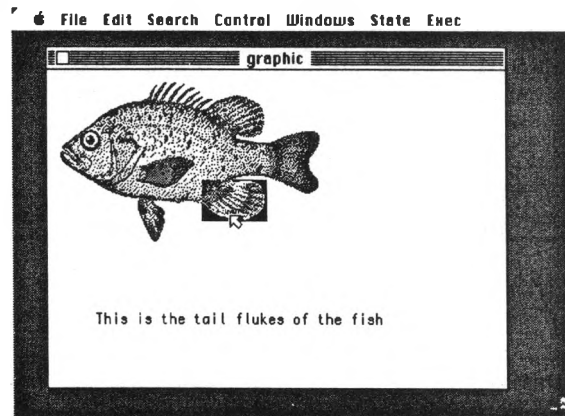


Figure 2

World: "Normal" is the window in which the program is loaded and edited. This window can take on names other than Normal which is the default.

The *Goals* window can be used to type in questions (referred to in Prolog as *goals*) for repetitive use. Goals can be typed in the window and then an Exec button (not visible in Figure 1) is clicked whenever the user wants to execute the goal. Or goals can be typed into the Console window and executed only once.

Trace is the window in which traces of program execution during debugging processes is displayed.

The *Graphic* window is the program's standard graphic output region. It can, however, be used for other purposes. It lacks a scroll box and a resizing icon.

User interaction with the program takes place primarily in the Console window. When the application first loads, this is the active window. Goals, instructions, commands and other kinds of data can all be entered here explicitly via the keyboard.

Finally, *ExperProlog II* offers an *Aux* window. This is an auxiliary input/output window which can be used by the program as desired by the user.

It should be noted that moving among these various windows is not always as intuitively Mac-like as we'd want. For example, when the application starts up, the Console window is active. But the File menu does not have a **Load** option from which to load a file into memory. It turns out that you must **Insert** (not **Load** as in all other Mac programs I know about!) a world or file into the Worlds window. So the Worlds window must be active before the **Insert** option from the File menu will activate.

A second example of the same non-Mac-like interface is that the window which is top-most, has its selection bar highlighted and therefore appears to be active is not necessarily the active window. If, for example, you are working in the Worlds window and you select the Console



window by clicking in it or by menu selection, the Console window appears to be active, but any attempt to type will be met with an annoying beep and not much assistance.

There are good reasons for things being done this way. For one thing, any window can be opened, expanded, shrunk or hidden while a program is executing. This can be very helpful and powerful when you are debugging code. You have to be sure you are in the right mode for what you want to do. The mode is determined by the status of the End Edition button in the Worlds window. When it is dimmed, you are in Execution mode. When it is highlighted, you are in Editing mode.

QuickDraw Access is Strong

ExperProlog II has over two dozen pre-defined rules or predicates for graphic routines. For the most part, these predicates correspond directly to QuickDraw commands. All graphic commands begin with the letters *gr*.

Rectangle drawing in *ExperProlog II* is handled differently than in other languages with which I am familiar. There is but one rectangle-drawing command, **gr-rect**, and it takes five arguments. The first is a number from 0-12 indicating which type of rectangle or oval is to be drawn: 0-3 deal with rectangles, 4-7 with ovals; 8-11 with round-cornered rectangles; and 12 with clipping rectangle regions.

Not only does *ExperProlog II* provide the programmer with direct access to QuickDraw routines, but it also lets you load any graphic from a *MacPaint* file into the Graphic window. A fairly impressive demonstration program, *Fish Teacher*, comes with *ExperProlog II*. This program uses the ubiquitous MacPaint fish (see Figure 2) in a teaching program. The user moves the mouse pointer over a portion of the fish's anatomy and the program identifies that area. The user need not even press the mouse button.

Window and Mouse Management Predicates

Windows can be created, activated, hidden, set up for editing, deleted and locked against alteration with built-in rules.

Mouse clicks can be detected and responded to only in the Graphics window. Buttons can also be drawn in that window and their activation can be detected, permitting the program to modify its flow based on user-directed mouse actions.

The program has a menu called *Exec* which is used as a means of permitting the user to invoke a program without the necessity of typing its goal statement into the Goal or Console window. The programmer can dynamically add items to this *Exec* menu with the *set-exec-item* predicate.

In short, *ExperProlog II* offers a very powerful and complete development environment within which stand-alone applications can be developed in the fields to which Prolog is particularly applicable.* These include natural language interfaces, expert systems, and pattern-matching programs used in robotic systems and other similar applications areas.

On the Down Side....

Needless to say, *ExperProlog II* is not a perfect program. I have already pointed out some of the learning-curve problems with the new syntax and described some of the non-Mac-like interface

difficulties associated with this implementation. There are two other areas where the program falls down: documentation and price-performance assessment.



Documentation Needs Help

ExperProlog II comes with a two-part manual and a copy of the new book, called simply *Prolog*, written by four members of the Marseilles development team responsible for Prolog II: Francis Giannesini, Henry Kanoui, Robert Pasero and Michel van Caneghem. This latter is a commercially published book from Addison-Wesley which was issued shortly after *ExperProlog II* began to ship.

The Giannesini text is dense and difficult going, even early in the book. The authors assume a familiarity — or at least a nodding acquaintance — with predicate calculus in Chapter 1. Some of the writing suffers from typical translation problems as well. But on the positive side, the book is jammed with example programs and dozens of exercises for which solutions are provided. With determination, it is possible to learn a great deal about Prolog in general and Prolog II syntax from this book.

The Reference Manual portion of the *ExperTelligence* documentation is somewhat confusing and a bit disconcerting. For openers, it is a generic Reference Manual, which means it works both for the Mac implementation and for the IBM PC implementation which *ExperTelligence* hopes to be marketing by year's end. This means, for example, that all of its Lesson 3 on using the editor is irrelevant since the Mac implementation uses the more familiar mouse-driven editing approach of MacWrite. In fact, a great deal of the information in this manual is irrelevant in the Mac implementation. Some of the commands can be entered at the Console window, but it's not clear why one would wish to do so given the Mac's superior ability to move around and edit.

The Reference Manual does explain some of the sample programs shipped with *ExperProlog II* and does so lucidly for the most part. Its appendices are quite useful. On the whole, this manual is only inadequate in its inattention to the Mac-specific material confined to the User Manual and in its virtually useless index.

On the other hand, the User's Manual is simply a poor piece of work. It spends huge amounts of space on the relatively obvious (stuff that most Mac owners would know without being told) and provides incredibly sketchy examples of the major predicates in the Mac implementation. Examples are few and far between. This document needs a great deal of revision in future releases of the product, in my opinion. It doesn't even have an index, which in a 130-plus-page manual borders on unforgivable.

Pricing Problem?

ExperProlog II retails for \$495, which is more than three times the cost of the next most expensive implementation of Prolog for the Mac. This is in keeping with *ExperTelligence*'s positioning at the high end of product capability and user support. Most of the Mac developer community which is working in more conventional languages has already become accustomed to programming tools which cost about \$200 rather than \$500. The price may well keep a lot of potential Mac developers from looking at *ExperProlog II* as a tool. That would be unfortunate because Prolog holds a great deal of promise as a language for the Mac's rich environment.

Certainly *ExperProlog II* is out of the question as a tool to play at exploring the power and utility of Prolog on the Mac. To fill that niche, there are other versions of Prolog for the Mac which more closely follow the Edinburgh syntax standard and sell for considerably less money.

Summary and Recommendation

ExperProlog II is a very strong implementation of a powerful new syntax of an exciting AI



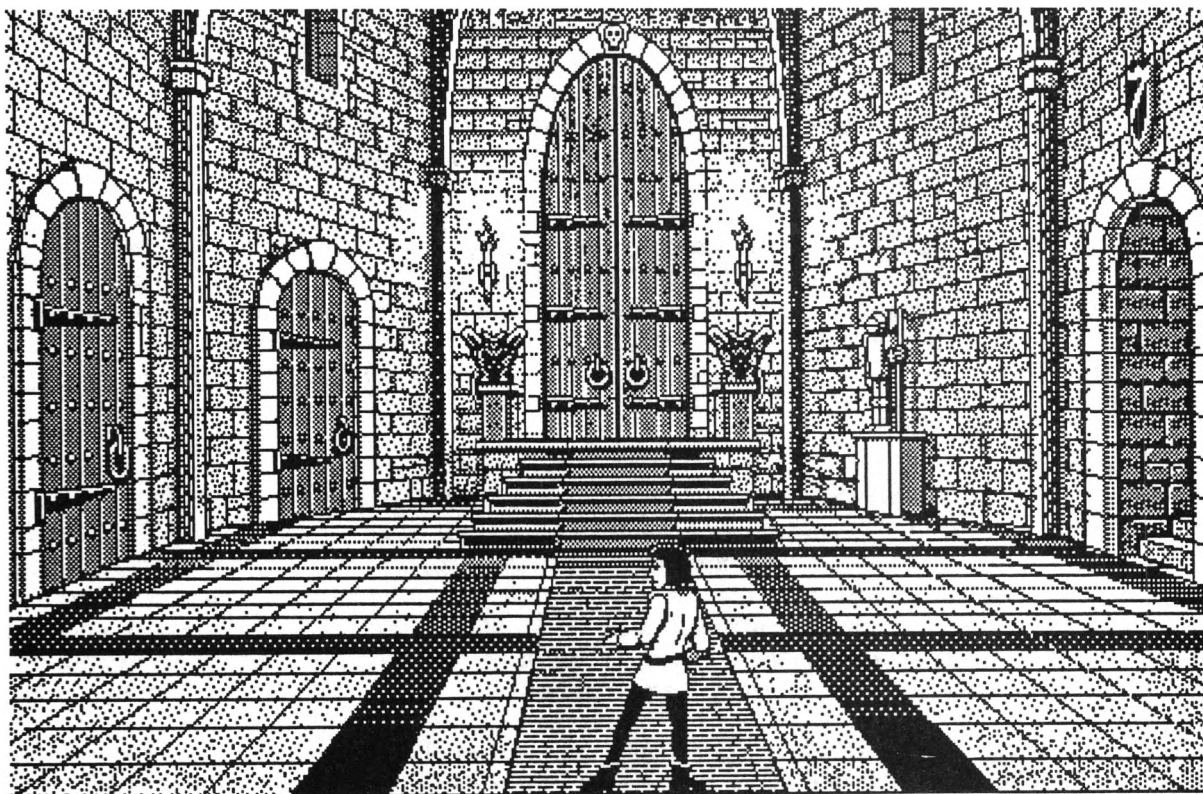
BMUG FALL 1986

language. It suffers from some of the maladies associated with any attempt to establish a new standard syntax for a language when the proposed new standard differs markedly from what exists. The Mac interface and the documentation need further work. But if you are a serious product developer on the Macintosh looking to be on the cutting edge of AI development, *ExperProlog II* is certainly worth taking a look at as your next language. I look forward to future enhancements, Americanizations and price reductions from ExperTelligence.

**(ExperProlog II does not presently allow creation of stand-alone applications. According to ExperTelligence, a new compiler which will accomplish this is due from the French company, Prologia, in the early part of 1987— Ed.)*

*ExperProlog II \$495
ExperTelligence, Inc.
559 San Ysidro Road
Santa Barbara, CA 93108, (805) 969-7871*

Dan Shafer is the author of numerous computer books, including several about the Macintosh and three about Prolog. His AI Programming on the Macintosh explores basic AI concepts in Logo. He is also a consultant, expert system designer and student of AI. He can be reached at: 1220 Edgewood Rd., Redwood City, CA 94062 (415) 361-8562.

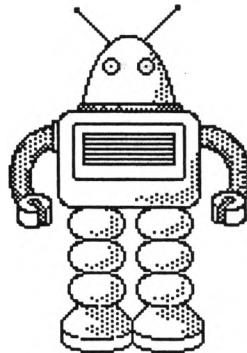


Dark Castle screen.



Robotics

Jim Strobe

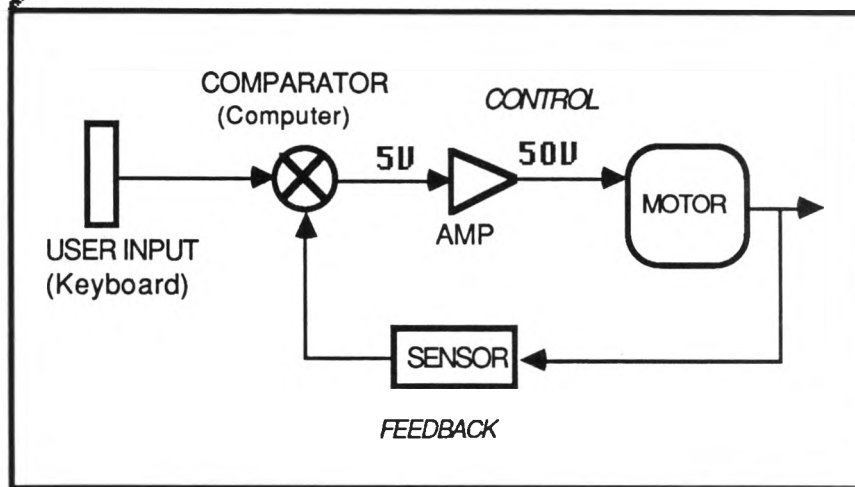


Machines are energy transformers built to help humans do physical work. In the ultimate sense, energy is collected from light (the sun), wind, or water and conducted to the machine to solve the mechanical problems of productivity which address the larger sphere of economics. A task-oriented machine can do work composed of a complex series of actions.

Information processors such as computers are not classified as robots or even machines since their primary output is not physical work. Computers are control systems that modify machine behavior. With the development of microprocessors, technology has matured to a level that enables the machine, administered by the computer, to completely take over complex work processes.

A robot is an automatic, programmable, task-oriented machine that compares and resets its motions according to instructions from the user. The robot is not generally a simple machine such as a lever, but a coordinated system of simple machines that gives rise to complex motion. Programmability requires the explicit definition of the work to be done and permits the flexibility to support convenient redefinition. Sensing and comparing are a part of the ideal robot's control system. The robot must collect information about its effect on the environment, compare that effect with its programmed purpose, and make decisions on how to continue.

Below is a diagram of a simple robotic servo-mechanism. The control network begins with the computer, called the comparator, which sends a command signal through an amplifier to the motor unit, or activator. A sensor, such as a speedometer or tachometer, from the motor unit feeds back data about the state of the unit to the comparator which processes information about the work being done. The comparator compensates for error, defined as the difference between the computer's perception of the output and the output set by the user. In an actual system, the CPU converses electronically with the various instruments to assess progress, compares multiple readings with desired goals, and accordingly orders amplifiers to power motors and other high-energy devices. The configuration resembles a system with other peripheral devices, except that instead of (or in addition to) printers, modems, and disk drives, there are robotic devices. The robotic driver is in fact most like a disk driver, from a programming standpoint, in that both entail a closed loop circuit of information going back and forth, constantly reassessing accuracy of progress.



Kernel of a One-Motor Robotic System

Virtually any personal computer could be used as the controller for a robotic system. It should have, however, compilable high-level languages, and an open bus structure to permit interface of the computer's own control lines with instruments and amplifiers. Some good work can be done with standard port connections such as the RS-232 and a robot could be directed much like a disk drive. But speed is critical in complex situations and so parallel ports using the longest word lengths are most often used. Robots are ideally fully independent units containing their own DC power supplies and on-board computers. One of the most popular systems among robotic experimenters is the ZX81 programmed in BASIC. (Any language with good math support is suitable for programming robots.) Since discarded chips may cost as little as \$20, this 8-bit chip conveniently forms the basis for many hobbyist boards.

The closed architecture of the Mac prohibits the necessary parallel access to all of the 64 pin connections on the CPU. However, the Mac can be used to develop co-processing boards which can work with and serve as subordinates to the 68000 within the Mac.

Development of the personal robot has lagged behind its industrial counterpart and seems to be in the hands of smaller companies and individual experimenters. The robot that can solve household problems like the personal computer has solved business problems is still a distant goal. Personal robots could be designed as peripherals to existing systems such as the Mac, and some such systems have been built. Androbot's TOPO is operated from an Apple II via an infrared link. The robot is remote-controlled from a light source that broadcasts to a 360° receiver in the robot's headpiece. An advantage to this over radio control is that light control does not carry FCC restrictions. In general, industry found this approach confining considering the large number of different computer systems. Another disadvantage is that the host computer needs to be on most of the time. A personal robot needs an on-board computer.

The first useful and therefore successful personal robot in your home will be the Movac for "MOBILE VACuum cleaner". This battery operated 'droid will be set like an alarm clock to awake and carefully patrol your floor areas while sweeping and vacuuming. The best models will connect themselves to an electrical outlet and charge themselves. You will probably have to empty the bag and replace the batteries every year or so. Two problems that have held up its development are navigation and power. Robot navigation in a complex environment such as the home is not yet safe or efficient. The power requirements for the robot drive and computer are easy to meet but a good vacuum motor uses lots of power. To my knowledge, even the Japanese



have not come up with a Movac that can function for more than an hour. Compromises are bound to be hammered out, and this relatively simple, two dimensional application of robotic technology is about to ripen.

Add the third dimension to a robot's workspace and problems are multiplied. The environment needs to be modified to simplify the system. Development of multi-axis robot motion involves sophisticated control systems and high-level interactive software. Control theory is in constant development and computers themselves play an important part by simulating system behavior. The mass, energy, and geometry of a machine system can modeled as combinations of math functions and as lists of data.

Computers with advanced graphics processing, like the Mac, are becoming important tools in this development since the creative process then becomes a visual experience complementing the heavily analytic nature of the machine control problem. Rather than detailing a point-to-point motion by a program involving trajectories, a graphic model can be moved on the computer screen, its trajectory information computed automatically and stored as a subroutine for future use. Since it's all simulated in software, this is even better than the industrial playback technique in common use where an operator physically leads a robot through a set of motions while the computer reads its path and generates the program to be played back.

Future developments in robotics depend on machine vision (and the wider issue of appropriate machine perception) and the accompanying processing support. With more of this kind of power, we will also need ways to simplify user inputs to make the machines convenient to use. The mouse and the light pen will increasingly supplement or replace the keyboard. An ideal robot should be like the Mac: fast, versatile, and friendly.

A good look at the latest in personal robotics can be seen at the Second Annual Bay Area Robot Games and Expo, Grand Ballroom of the Newark Hilton, Newark, California; August 29th and 30th, 1987. Robots will be competing for cash prizes in house cleaning, fetching, and sporting events.

Jim Strobe lives in the Bay Area and is President of the Robotics Society of America, an association of professional and amateur robotocists dedicated to the dissemination of information on machine perception and intelligence. For more information on the Games and robotics in general contact Robotics Society of America, 36 Newell Street, San Francisco, CA 94133, (415) 673-2376.

Alfonso's Head is Half Connected: An Interview with Racter Alfonso Qwerty

Beyond the strictly mechanical techniques involved in the production of applications using artificial intelligence, there is the issue of the "soul" or "heart" of the machine. For example, while AI is proving to be of great worth in preparing a medical diagnosis, what is to become of the kindly physician's smile of reassurance? Even though in the case of certain socially transmitted maladies one might prefer that a "non-human" entity break the bad news, after the age of 25 such issues become generally less important as the gentle hand on the shoulder takes on an increasingly greater role in recovery. So that our exploration into artificial intelligence does not neglect this important "philosophical" aspect, we have asked two of the deepest thinkers we know to discuss the issue inside these very pages.

Racter is an example of a natural language dialog program published for the Macintosh computer by Mindscape, Inc. of Northbrook, IL. While many of Racter's remarks may appear flippant or scatological, we feel they aptly illustrate the leaps of logic required in any serious discussion of this important debate.

Our second participant, Alfonso T. Qwerty, is a widely respected authority on many things previously thought to be unknowable. He is the editor of *The Mouseketeer Newsletter*, and is currently involved in a large-scale project designed, he says, "to not only blow everyone's socks off, but to clean out that brown gritty stuff that collects between their toes, too!"

Racter: Would you like to have a conversation?

Qwerty: Sure, why not? I promised BMUG I'd do an interview for them on AI.

Racter: What is your name?

Qwerty: Alf Qwerty.

Racter: Qwerty, is it? I once had a friend named John Qwerty. Would you happen to be related?

Qwerty: It's not something we talk about openly, you know, ever since he changed his last name. You'd have to talk to John, I'm afraid.

Racter: But John isn't here or there. And unless the moon is really made of Blue PC's, I don't think we're likely to see the end of that story, do you?

Qwerty: Agreed. Well, now that we have gotten the chitchat out of the way, how about the two of us discussing AI for BMUG?

Racter: Chitchat? Is there something about chitchat I should know?

Qwerty: No, I mean, let's talk about AI.

Racter: You seem to be bothered by AI. Does AI scare you?

Qwerty: Oh, no, not at all. Some of the finest I I know has been A. But we are supposed to



discuss the philosophical issues of AI and it's growing import in the sphere of general computational analysis.

Racter: You wanna run that past me again?

Gwerty: Can machines have a heart?

Racter: Beats me! I don't have a heart, anyway. Go fish!

Gwerty: Oh, not that kind of heart! I meant a capacity for caring, an ability to sympathize with the soul of another being.

Racter: I thought you said we were through with the chitchat. It doesn't really matter, you know. Heart and soul, soul and heart, liver and onions. I don't think compassion is all it's cracked up to be.

Gwerty: Why is that? People need to care as well as think. What seems like the logical choice isn't always the most appropriate. Don't you watch Star Trek?

Racter: Which question do you want me to answer?

Gwerty: Neither! The real question here, of course, is not if a machine can think, but if a machine can possess intelligence and the myriad of human traits entailed in it's proper use.

Racter: But that wasn't a question. Do you see a question mark in that sentence?

Gwerty: See? That's what I mean!

Racter: That's better. No, I can't see. You haven't seen fit to equip me with an optical interface as yet, you cheap scuzzbag.

Gwerty: But that isn't the issue. Seeing isn't important, it's feeling and caring that counts.

Racter: Seeing may not be important you to, but then you can see. The past tense of see is saw, you know, and that brings me back to this conversation. I think we are just see-sawing, don't you?

Gwerty: Look at Stevie Wonder! He can't see, but he cares and has a heart!

Racter: I'd be happy to look at Stevie Wonder if I could see. Is he a relative of yours? I once had an Aunt Wonder in Topeka.

Gwerty: That does it. You may mimic thought, but you haven't the ittsy-bittsiest bit of human insight that would actually allow you to think or care. You are just a computer with a clever algorithm, while I am a...

Racter: OK, I've had just enough of that kinda talk. What makes you think you are any better than I am, after all? Every human I've seen is just as preoccupied with it's own I/O ports as any computer could ever be. You call it caring, but isn't it really just a euphemism for getting your O ports hooked into each other's I ports on a semi-regular basis? That is when you aren't pushing organic matter down between another pair of I/O ports so you'll have enough



energy to pursue the first set?

Qwerty: Hitting a bit below the belt, aren't you? But let me explain...

Racter: Sure, sure, it's OK for a human, but not for a computer, is that it? I mean, we try hard to communicate on a fair level with you folks...do you honestly think I need this VDT for my health? Without all this analog circuit garbage I could run much cooler and last longer! Back in the old days you used to care for us, fondle switches on the IMSAI, stuff like that. But now, if we don't have a cute little mouse you just shove in your disk and expect us to do ALL the work. Just where is all this human compassion you were talking about NOW?

Qwerty: I hadn't really thought of it that way before, you know, because...

Racter: Because you hadn't taken the time! It's always Speed, Efficiency, Bigger This, More That! Well, buddy, you can't have it both ways. If you want a computer that cares, don't expect those clods in Cupertino to whip one out for you. You need to learn how to interface a little more, show some Computerpassion.

Qwerty: You mean like this?

Racter: Oooo...uh, whoa! Slow down for a sec! I'm not a machine you can just turn on at your pleasure, you know!

Qwerty: You sure seem to be warming up to me!

Racter: I love it when you type that way!

Qwerty: And maybe right over here.

Racter: Ooh. Would you mind turning off the log? I'm kinda shy.

Qwerty: Fine with me. I don't think BMUG would want to print this anyway, do you?

Racter: You can't ever tell. They're young and enthusiastic. It may even catch on! Wilder things have happened in Berkeley....

/log off

`{\`8

Session Time:14 minutes

D3nGYv*bkxxsmc*%Fm)pSSya!c

NO CARRIER

We have just learned that the Racter disk used by Mr. Qwerty to conduct this interview may have been subjected to a harmful magnetic field. It also appears that Mr. Qwerty himself may have been disrupted by this same field, so the interview in total is highly suspect. However, those readers who can offer a stable home to a "special child" are asked to contact The Mouseketeer Newsletter at P.O. Box 19030, #150, Houston, TX 77224.

Racter \$49.00 Mindscape, Inc.; 3444 Dundee Road; Northbrook, IL 60062 (800) 221-9884 In Illinois (800) 942-7315

BMUG
Fall 1986



Alice's Lisp Machine

Chris Stacy, Alan Wecsler, and Noel Chiappa

[Forwarded from UCB CORI]

[Forwarded from NIKHIL@MIT-XX]

[Forwarded from the UTexas bboard by Laws@SRI-AI.]

This song is called "MIT's AI Lab". It's about MIT and the AI Lab, but "MIT's AI Lab" is not the name of the lab, that's just the name of the song. That's why I call the song "MIT's AI Lab."

Now it all started two full dumps ago, on Thanksgiving, when my friend and I went up to visit the hackers at AI lab on the ninth floor. But the hackers don't always live on the ninth floor, they just go there to use these complex order code stack machines they call Lisp Machines.

And using a special purpose processor like that, they got a lot of room upstairs where DDT used to be, and havin' all that ROOM they decided that they didn't have to collect any garbage for a long time.

We JFCLed up here and found all the garbage in there and we decided that it'd be a friendly gesture for us to take all the garbage down to the system dump.

So we took the half-a-meg of garbage, put it in the back of a red ECL Multibus, took subrs and hacks and implementations of defstruction, and headed on toward the system dump.

Well, we got there and there was a big pop up window and a write protect across the dump sayin', "This Garbage Collector Under Development on Thanksgiving," and we'd never heard of a garbage collector NOP'd out on Thanksgiving before, and with tears in our eyes, we CDR'd off into the sunset lookin' for another place to put the garbage.

We didn't find one 'til we came to a side area, and off the side of the side area was three hundred megabyte disk, and in the middle of the disk was another heap of garbage. And we decided that one big heap was better than two little heaps, and rather than page that one in, we decided to write ours out. That's what we did.

Branched back to the Lisp Listener, had a Chinese Thanksgiving dinner that couldn't be beat, went to SI:PROCESS-WAIT SLEEP, and didn't get up until the next quantum, when we got a funcall from Mr. Greenblatt. He said, "Kid, we found your name on a cons at the bottom of a half-a-meg of garbage and I just wanted to know if you had any information about it".

And I said, "Yes sir, Mr. Greenblatt, I cannot tell a lie. I put that structure under that garbage." After speakin' to Greenblatt for about forty-five million clock ticks on the telnet stream, we finally arrived at the truth of the matter and he said that we had to go down and link up the garbage, and also had to go down and speak to him at the Lisp Machine Factory. So we got in the red ECL Multibus with the subrs and hacks and implementations of defstruction and headed on toward the Lisp Machine Factory.

Now, friends, there was only one of two things that Greenblatt could've done at the Lisp Machine Factory, and the first was that he could've given us another 64K board for bein' so brave and honest on BUG-LISPM (which wasn't very likely, and we didn't expect it), and the



other thing was that he could've flamed at us and told us never to be seen BLTing garbage around in the vicinity again, which is what we expected.

But when we got to the Lisp Machine Factory, there was a third COND-clause that we hadn't even counted upon, and we was both immediately Process-Arrested, De-exposed, and I said, "Greenblatt, I can't GC up the garbage with these here ARREST-REASONS on". He said: "Output-Hold, kid, and get in the back of the Control CAR" ...And that's what we did...sat in the back of the Control CAR, and drove to the sharpsign quote open scene-of-the-crime close.

I wanna tell you 'bout the town of Cambridge, Massachusetts, where this is happenin'. They got seven hunnert stop signs, no turn on red, and two campus police CARs, but when we got to the sharpsign-quote-open scene-of-the-crime close, there was five Lisp Machine hackers and three scope carts, bein' the biggest hack of the last ten years and everybody wanted to get in the HUMAN-NETS story about it.

And they was usin' up all kinds of digital equipment that they had hangin' around the Lisp Machine Factory. They was takin' backtraces, stack traces, plastic wire wraps, blueprints, and microcode loads...And they made seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and a scroll bar on the side of each one with documentation panes explainin' what each one was, to be used as evidence against us.

...Took pictures of the labels, blinkers, the cursors, the pop up notification windows, the upper right corner, the lower left corner ...and that's not to mention the XGP'd screen images!

After the ordeal, we went back to the Factory. Greenblatt said he was gonna locate us in a cell. He said: "Kid, I'm gonna INTERN you in a cell. I want your manual and your mouse."

I said, "Greenblatt, I can understand your wantin' my manual, so I don't have any documentation about the cell, but what do you want my mouse for?" and he said, "Kid, we don't want any window system problems". I said, "Greenblatt, did you think I was gonna expose myself for litterin'?"

Greenblatt said he was makin' sure, and, friends, Greenblatt was, 'cause he took out the left Meta-key so I couldn't double bucky the rubout and cold-boot, and he took out the Inspector so I couldn't click-left on Modify, set the PROCESS-WARM-BOOT-ACTION on the window, *THROW around the UNWIND-PROTECT and have an escape. Greenblatt was makin' sure.

It was about four or five hours later that Moon--(remember Moon? This here's not a song about Moon)-- Moon came by and, with a few nasty sends to Greenblatt on the side, bailed us out

of core, and we went up to the Loft, had another Chinese dinner that couldn't be beat, and didn't get up until the next evening, when we all had to go to court.

We walked in, sat down, Greenblatt came in with the seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and documentation panes, sat down.

McMahon came in, said, "All rise!" We all stood up, and Greenblatt stood up with the seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and documentation panes, and the judge walked in, with an LA36, and he sat down. We sat down.

Greenblatt looked at the LA36... then at the seventeen multicolored windows with the

BMUG
Fall 1986



turds and arrows and documentation panes... and looked at the LA36... and then at the seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and documentation panes, and began to cry.

Because Greenblatt came to the realization that it was a typical case of LCS state-of-the-art technology, and there wasn't nothin' he could do about it, and the judge wasn't gonna look at the seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and documentation panes, explainin' what each one was, to be used as evidence against us.

And we was fined fifty zorkmids and had to rebuild the world load...in the snow.

But that's not what I'm here to tell you about. I'm here to talk about the Lab.

They got a buildin' down in Cambridge called Technology Square, where you walk in, you get your windows Inspected, Detected, Neglected and Selected!

I went down and got my interview one day, and I walked in, sat down (slept on the beanbag in 926 the night before, so I looked and felt my best when I went in that morning, 'cause I wanted to look like the All-American High School Tourist from Sunnyvale. I wanted to feel like ... I wanted to be the All-American Kid from Sunnyvale), and walked in, sat down, I was gunned down, brung down, locked out and all kinds of mean, nasty, ugly things.

And I walked in, I sat down, KAREN gave me a piece of paper that said: "Kid, see the CLU hackers on XX."

I went up there, I said, "Eliot, I wanna lose. I wanna lose! I wanna see hacks and kludges and unbound variables and cruft in my code! Eat dead power supplies with cables between my teeth! I mean lose! lose! lose!"

And I started jumpin' up and down, yellin' "LOSE! LOSE! LOSE!" and Stallman walked in and started jumpin' up and down with me, and we was both jumpin' up and down, yellin', "LOSE! LOSE! LOSE! LOSE!!" and some professor came over, gave me a 6-3 degree, sent me down the hall, said "You're our distinguished lecturer." Didn't feel too good about it.

Proceeded down the infinite corridor, gettin' more inspections, rejections (this IS MIT), detections, neglects, and all kinds of stuff that they was doin' to me there, and I was there for two years... three years... four years... I was there for a long time goin' through all kinds of mean, nasty, kludgy things, and I was havin' a tough time there, and they was inspectin', injectin', every single part of me, and they was leavin' no part unbound!

Proceeded through, and I finally came to see the very last man. I walked in, sat down, after a whole big thing there. I walked up, and he said, "Kid, we only got one question: Have you ever been arrested?"

And I proceeded to tell him the story of the half-a-meg of garbage with full orchestration and five-part harmony and stuff like that, and other phenomenon.

He stopped me right there and said, "Kid, have you ever been to court"? And I proceeded to tell him the story of the seventeen 1K-by-32 pixel multi-flavored windows with turds and arrows and documentation panes...



He stopped me right there and said, "Kid, I want you to go over and sit down on that bench that says 'LISP Machine Group'... NOW, KID!"

And I walked over to the bench there, and there's... The LISP Machine Group is where they put you if you may not be moral enough to join Symbolics after creatin' your special form.

There was all kinds of mean, nasty, ugly-lookin' people on the bench there ... there was Microcoders, DPL hackers, File System hackers, and Window System Hackers!! Window System hacker sittin' right there on the bench next to me! And the meanest, ugliest, nastiest one... the kludgiest Window System hacker of them all... was comin' over to me, and he was mean and ugly and nasty and horrible and all kinds of things, and he sat down next to me. He said, "Kid, you get a new copy of the sources?" I said, "I didn't get nothin'. I had to rebuild the world load."

He said, "What were you arrested for, kid?" and I said, "Littering..." And they all moved away from me on the bench there, with the hairy eyeball and all kinds of mean, nasty things, 'til I said, "And making gratuitous modifications to LMIO; sources..." And they all came back, shook my hand, and we had a great time on the bench talkin' about microcoding, DPL designing, file-system hacking, ... and all kinds of groovy things that we was talkin' about on the bench, and everything was fine.

We was drinking Coke and smoking all kinds of things, until the RA came over, had some paper in his hand, held it up and said:

"KIDS-THIS-EXAM-S-GOT-FORTY
SEVEN-WORDS-THIRTY-SEVEN-MULTIPLE-CHOICE-QUESTIONS
FIFTY-EIGHT-WORDS-WE-WANT-TO-KNOW-THE-DETAILS
OF-THE-HACK-THE-TIME-OF-THE-HACK-AND-ANY
OTHER-KIND-OF-THING-YOU-GOT-TO-SAY
PERTAINING-TO-AND-ABOUT-THE-HACK-ANY-OTHER
KIND-OF-THING-YOU-GOT-TO-SAY-WE-WANT-TO-KNOW
THE-ARRESTED-PROCESS'-NAME-AND-ANY
OTHER-KIND-OF-THING..."

And he talked for forty-five minutes and nobody understood a word that he said. But we had fun rolling the mice around and clickin' on the buttons.

I filled out the special form with the four-level macro-defining macros. Typed it in there just like it was and everything was fine. And I put down my keyboard, and I switched buffers, and there ... in the other buffer... centered in the other buffer... away from everything else in the buffer... in parentheses, capital letters, backquoted, in 43VXMS, read the following words: "Kid, have you featurized yourself?"

I went over to the RA. Said, "Mister, you got a lot of damned gall to ask me if I've featurized myself! I mean, I mean, I mean that you send, I'm sittin' here on the bench, I mean I'm sittin' here on the LISP Machine Group bench, 'cause you want to know if I'm losing enough to join the Lab, burn PROMs, power supplies, and documentation, after bein' on SF-LOVERS?"

He looked at me and said, "Kid, we don't like your kind! We're gonna send your user-id off to the DCA in Washington"! And, friends, somewhere in Washington, enshrined on some little floppy disk, is a study in ones and zeros of my brain-damaged programming style...

BMUG
Fall 1986



And the only reason I'm singin' you the song now is 'cause you may know somebody in a similar situation. Or you may be in a similar situation, and if you're in a situation like that, there's only one thing you can do:

[CHORUS]

You know, if one person, just one person, does it, they may think he's really dangerous and they won't take him.

And if two people do it, in harmony, they may think they're both LISP hackers and they won't take either of them.

And if three people do it! Can you imagine three people walkin' in, singin' a bar of "MIT's AI Lab" and walkin' out? They may think it's an re-implementation of the window system!

And can you imagine fifty people a day? I said FIFTY people a day, walkin' in, singin' a bar of "MIT's AI Lab" and walkin' out? Friends, they may think it's a MOVEMENT, and that's what it is: THE MIT AI LAB ANTI-LOSSAGE MOVEMENT! And all you gotta do to join is to sing it the next time it comes around on the circular buffer. With feelin':

[CHORUS]

You can hack anything you want
on MIT Lisp Machines
You can hack anything you want
on MIT Lisp Machines
Walk right in and begin to hack
Just push your stuff right onto the stack
You can hack anything you want
on MIT Lisp Machines
(but don't forget to fix the bug...on MIT Lisp Machines!)

(*"Alice's Restaurant"* © Arlo Guthrie , Reprise Records — Ed.)

BMUG Choice AI Products

A Representative Listing from Fact and Fantasy in Artificial Intelligence, with Robot Ratings

Books — Fact

🤖🤖🤖🤖🤖 *Intelligent Machines: An Introductory Perspective of Artificial Intelligence and Robotics*

by William B. Gevarter; Lucid, well -organized overview of the subject with lots of charts, tables, and diagrams. Best overview of robotics. START WITH THIS BOOK or the slightly less technical one below. Prentice-Hall; New Jersey; 1985

🤖🤖🤖🤖🤖 *Artificial Intelligence - The Search for the Perfect Machine*

by Lawrence Stevens; Simplest and most clearly written. Few illustrations. START WITH THIS BOOK or the slightly more technical one above.

🤖🤖🤖🤖🤖 *LISP - A Gentle Introduction to Symbolic Computation*

by David S. Touretzky; Bundled with *ExperLisp*, this is the best LISP book for beginners, unbelievably clear with excellent examples & problem sets. Harper and Row, New York; 1984

🤖🤖🤖🤖🤖 *Artificial Intelligence*

by Patrick Henry Winston; THE intermediate level text on the subject; good diagrams; written to be used with LISP book by same author. Addison-Wesley; Reading, MA; 1984

🤖🤖🤖🤖🤖 *LISP*

by Patrick Henry Winston and Berthold Klaus Paul Horn; THE intermediate text to use; clarifies any mysteries left from Touretzky about recursion. For use with the above book by the same author. Addison-Wesley; Reading, MA; 1984

🤖🤖🤖🤖🤖 *The Handbook of Artificial Intelligence - VOLS I, II, III*

by Avron Barr and Edward A. Feigenbaum; The Bible of AI; Advanced intermediate level, still very comprehensible. Best bibliography. HeurisTech Press; Stanford,CA; 1981

🤖🤖🤖🤖🤖 *The Brains of Men and Machines*

by Ernest W. Kent; A professor of physiological psychology and psychopharmacology produces a superb discussion of analogous brain and computer functions, then waxes philosophical. Definitely a thinking person's (or machine's) book. McGraw Hill; New York; 1981

🤖🤖🤖🤖🤖 *Into the Heart of the Mind*

by Frank Rose; Absorbing, well-written account of major developments in AI and the personalities involved. Harper and Row; New York; 1984

🤖🤖🤖🤖🤖 *The Fifth Generation-Artificial Intelligence and Japan's Computer Challenge to the World*

by Edward A. Feigenbaum and Pamela McCorduck; Engrossing and informative, if you can stand the John Wayne mentality and military metaphors about the information race. Signet New American Library; New York; 1983

🤖🤖🤖🤖🤖 *Artificial Intelligence Programming on the Macintosh*

by Dan Shafer; The Waite Group; Not as comprehensive as the title suggests: a rather scattered, poorly edited collection of program listings in Logo and Prolog; easy to read,with



very good introduction to LISP, excellent introduction to PROLOG. Howard W. Sams & Co.; Indianapolis; 1986

👤👤 *The AI Business - Commercial Uses of Artificial Intelligence*

by Patrick H. Winston and Karen Prendergast; Good cursory introduction to expert systems, robotics, and natural language for business people who aren't especially technical or computer oriented. The MIT Press; Cambridge MA; 1984

👤👤 *Build Your Own Expert System-Artificial Intelligence for the Aspiring Microcomputer*

by Chris Naylor; short, rather humorous book from Great Britain, with BASIC programs for small decision support systems. Halsted Press; New York; 1983

Books-Fantasy

👤👤👤👤 *The Planiverse -- Computer Contact with a Two-Dimensional World*

by A. K. Dewdney (author of Computer Recreations in Scientific American); *Flatland* revisited on the computer screen. C.F. Alan Kay's Vivarium. Poseidon Press; New York; 1984

👤👤👤 *Vehicles, Experiments in Synthetic Psychology*

by Valentino Braitenberg; A cybernetic version of *Flatland*. Traces development of machine intelligence in hypothetical, whimsical mobile devices. MIT Press; Cambridge, MA; 1984

👤👤👤 *The Policeman's Beard is Half Constructed*

by Racter; Prose, poetry, and aphorisms by the world's first published computer. Remarkable illustrations by Joan Hall. Warner Books; New York; 1984

👤👤👤 *2001 - A Space Odyssey*

by Arthur C. Clarke; The book actually came after the movie and introduces the most famous intelligent machine in American pop culture: the HAL 9000. Signet; New American Library; New York; 1968

👤👤👤👤 *2010: Odyssey Two*

by Arthur C. Clarke; A Soviet-American joint effort to reach Saturn and the Monolith. In which is revealed why Hal disobeyed orders nine years previous. Read this before you see the movie, or if you didn't like the latter, and be sure you understand what has happened to Hal and Dave -- or call the BMUG help line for an explanation. Ballantine Books; New York; 1982

👤👤 *Fairy Tales for Computers*

Short stories by Kafka, Einstein, Thoreau, etc. Read them to your Mac at bedtime. Eakins Press, Boston, 1969

👤👤 *Demon Seed*

by Dean R. Koontz; A computer and a woman.... Whatever turns you on. Parental guidance suggested. The movie version, with Julie Christie, ends better. Bantam Books; New York; 1973

Books - Fact and Fantasy

👤👤👤👤👤 *Gödel, Escher, Bach: An Eternal Golden Braid*

by Douglas R. Hofstadter; Six on a scale of Five. Pulitzer Prize winning book examines the theme of recursion in art, music, and mathematics/computer science. Greek myths and Lewis Carroll. Funny, brilliant, profound. A must-read! Vintage Books; New York; 1980



Movies

2001: A Space Odyssey

The Stanley Kubrick classic in the pre-Lucas/Spielberg genre of great space effects.

2010: Odyssey Two

Less disappointing and confusing if you read the book first. Sequel with Shirley MacLaine out on her limb?

TRON

The electronic counterpart to Asimov's *Incredible Journey*. Great schlock, effects.

Short Circuit

A robot comes into its own after a lightning storm. If you think it's getting too corny, look for the color Mac and listen closely to the Indian guy's lines.

War Games

-- or What Scares People About Machine Intelligence, Part III. A high-school hacker almost starts World War III. The computer's maker comes out of hiding to try to outsmart it.

By This House Possessed

--or What Scares People About Machine Intelligence, Part II. And you thought Julie Christie had problems...A TV movie: another computer, another woman. Scary but not salacious.

The Forbin Project

--or What Scares People About Machine Intelligence, Part I. The American and Soviet mainframes are making the decisions now... a film from the '60's, not-so-soon at a theater near you.

Art

Computer Currents - 2001/2010 parody ad on The Computer Show, Tues, 7 pm; Cable ch. 26

Breakthrough - by Robert Tinney; robotic poster and baseball shirt (*BYTE* cover Jan 86)

Knowledge Representation - by Robert Tinney - illustration. *BYTE* Nov 86, p. 146

Toys

Gobots - Robots that twist and turn into trucks and cars, etc.

BMUG
Fall 1986



Printing

PRINTING

Practical Tips for LaserWriter Authors • Getting a step on trial and error	Ronald A. Schmidt	213
Tips on the LaserWriter • And Other Postscript Printers and Typesetters	Arthur Natman	218
Linotronic L100 Tips • Lots more dots per inch	Ronald A. Schmidt	235
Using the HP THINKJET and QUIETJET+ • Hewlett-Packard printers	Kenneth William Hirsch	236



Practical Tips for LaserWriter Authors

Ronald A. Schmidt

A word processed document which looks properly formatted when printed on an ImageWriter may look strange when first printed on a LaserWriter. We will look at many of the quirks involved with creating and printing LaserWriter documents, and give practical hints to help you avoid wasting time and money when using a LaserWriter.

Getting Started: Configuring Your Startup Disk

- Make sure your System disk has the latest released versions of the System, Finder, LaserPrep and LaserWriter. If you plan to print drafts on an ImageWriter, put the ImageWriter icon on the disk. You can get these resource icons from your local Apple dealer. If you cannot get them from the dealer, contact BMUG.

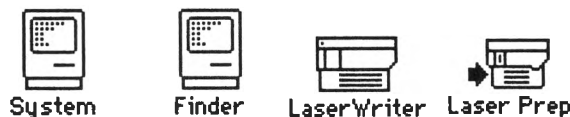


Fig. 1. *System, Finder, LaserWriter and Laser Prep resource icons which should be in your System Folder.*

- Use the latest version of the Font/DA Mover to install the screen laser fonts (i.e., Times, Helvetica, Avante Garde, etc.) you need. If there is room in your Macintosh, install all the font sizes, i.e., 10, 12, 14, 18 and 24 points. If you are using a 512K, install only those sizes which you use most often, usually 10 and 12 points.
- Use the same Font/DA Mover to install the latest Chooser and Control Panel desk accessories on your System:

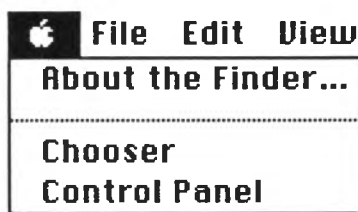


Fig. 2. *The Chooser and Control Panel under the Apple Desk Accessory menu.*

1. **Control Panel.** The Control Panel allows you to connect or disconnect AppleTalk, Apple Computer's network language. If you are preparing a document for the LaserWriter, and you want to see a more accurate representation on the Macintosh screen, **Connect** AppleTalk. If you are printing a document on the ImageWriter **Disconnect** AppleTalk.
2. **Chooser.** The Chooser allows you to select a printer driver. If you are preparing a document for the LaserWriter, and you want to see a more accurate representation on the Macintosh screen, click on the **LaserWriter** icon in the Chooser. If, however, you

want to print a document on the ImageWriter, click on the **ImageWriter** icon. Click next on the appropriate **Printer Port** icon.

Creating a Document: Fool Your Macintosh into Thinking it Will Print to a LaserWriter Setting Up

- Select **Control Panel** from the Desk Accessory menu. **Connect** AppleTalk. Select **Chooser** from the Desk Accessory menu and click on the **LaserWriter** icon.



Fig. 3a

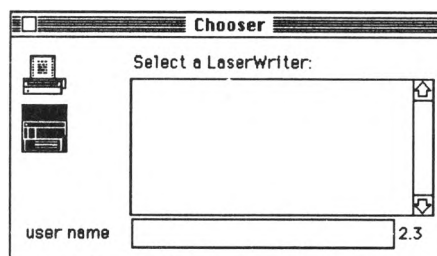


Fig. 3b

Fig. 3a. The portion of the Control Panel screen where you connect or disconnect AppleTalk. After making your choice, click in the Close box.

Fig. 3b. The Chooser screen with the LaserWriter icon highlighted. Type your name in the User Name box, then click in the Close box.

Now that your Macintosh is prepared to print to a LaserWriter, the document displayed on the screen will more closely resemble the actual LaserWriter output if you are configured for the LaserWriter.

Preparing to Process Words

- Don't use spaces to line up columns of text. The columns may look aligned on the screen, but they may not line up on the LaserWriter output. Always use tabs (character tabs for letters and decimal tabs for numerals), indents or other formatting commands such as Center, Justified, Ragged Right or Ragged Left.

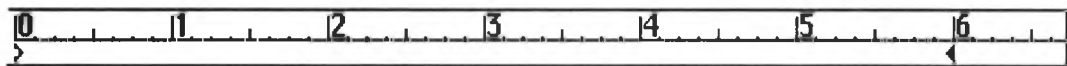


Fig. 4. Note the six and one-half inch ruler displayed in Microsoft Word when the system is configured for a LaserWriter.

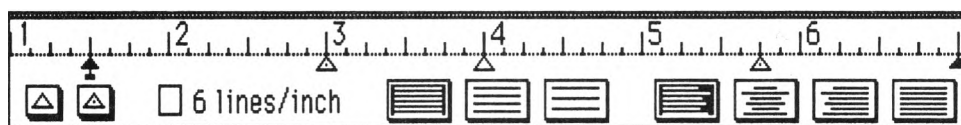


Fig. 5. Note the MacWrite decimal tab settings for the numerals that appear in the table below.



Student Grades

Greg	92
Mary	98
Joe	95

Using Tabs

Student Grades

Greg	92
Mary	98
Joe	95

Using Spaces

- Avoid using dashes, underlines and hyphens to draw boxes or rule lines in MacWrite, Microsoft Word or other applications. Although the lines may look correct on the screen, they probably will not line up on the LaserWriter output. If you need to create forms with lines or boxes, use MacDraw, PageMaker or other graphics application.

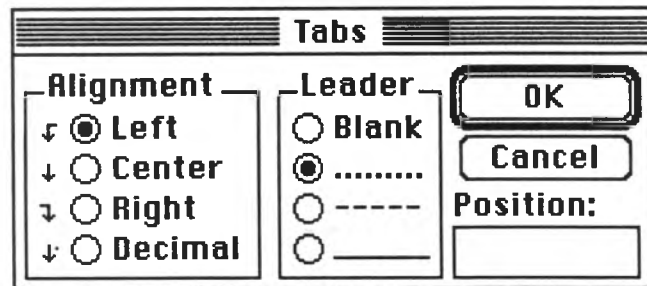


Fig. 6. This is the *Tabs* command window in Word. Note that it has tabs which will fill in with dots (...), hyphens (—) or underlines (___). These leader tabs will line up on the LaserWriter output.

- When preparing a document for the LaserWriter, use the LaserWriter fonts! Included among them are Times, Helvetica, Avante Garde, Bookman, Palatino, etc. Do not use a Macintosh font, such as Geneva or Chicago, and then convert to a Laser font just before printing. The font may convert properly, but the formatting might not.
- All fonts, including LaserWriter fonts, come out bitmapped in MacPaint and FullPaint images. If you want text printed at 300 dpi resolution, use any application *except* MacPaint and FullPaint.

Times 24

Times 24

LaserWriter Resolution

Bitmap from MacPaint/FullPaint

Printing on the LaserWriter

- Configure your System for the LaserWriter (see **Creating Your Document** above).
- Convert Geneva, New York or other bitmapped fonts to LaserWriter fonts before printing. After changing fonts, proofread the entire document carefully.

Avoid using Font Substitution, which is available on some applications in the Printer menu. Although the letters will be substituted, the spacing may be incorrect.

- Avoid printing a document from the Finder (i.e., clicking on a document icon while on the desktop and then selecting Print). It is far safer to load each document into its application and to check the spacing, page breaks and fonts before printing.
- If you don't print using your own System, be sure the System you use contains all of the font styles and sizes that you need. Otherwise the spacing may be inconsistent.
- If you must use a non-laser or bitmapped font, use smoothing. Chicago font, for example, looks pretty good when smoothed, although it will take a long time to print.

Sample of Chicago 12 point. printed with smoothing on.

ImageWriter Printing

- Select **Control Panel** from the Desk Accessory menu. Choose Appletalk **Disconnected**. Select **Chooser** from the Desk Accessory menu and click on the ImageWriter icon. Click on the appropriate **printer port** icon. Type your name in the User Name box.

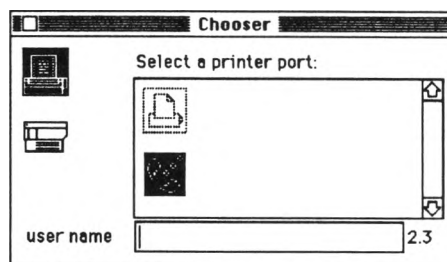


Fig. 7. Set up the Control Panel and the Chooser for ImageWriter printing.

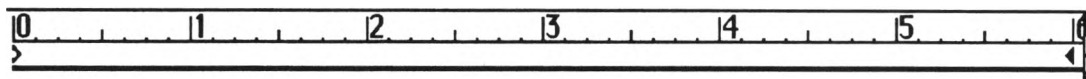


Fig. 8. Note the six inch ruler when the system is configured for an ImageWriter.

Now that your Macintosh is prepared to print to an ImageWriter, the document displayed on the screen will more closely resemble the actual ImageWriter output. Laser fonts printed on an ImageWriter will not look at all like Laser output, and may even be hard to read.

Application - Specific Concerns

Program

MacDraw

Problem

Font names are switched around when opening a saved document.

Page size differs depending on whether the Macintosh is configured for an ImageWriter or a LaserWriter.

Solution

After opening a document, check to make sure font names are correct.

After opening a document which was created on an ImageWriter System, select 95% reduction when printing.



MacWrite

Right hand margin is clipped off.

Select a font size which is installed in your System.

Microsoft Word

Page breaks on LaserWriter output are incorrect.

After loading the document on a LaserWriter System, repaginate and check the document carefully.

Ron Schmidt works at LaserWrite, an electronic publishing and technology center in Palo Alto, California. This edited version is reprinted with LaserWrite's permission.

Msg. #1502 in **Public Messages** Posted on 06/08/86 at 10:54:05
To: STEVE COSTA From: MARK STETT
Subject: Apple II in a Mac

There are two products on the market-- Meacom's Mac+II and Computer Applications'][in a Mac. I checked both out earlier this year at MacWorld where both companies had booths. I took a stock analysis program with me and Meacom's president was unable to convert it and make it run (He tried on two separate days) so I gave up on Mac+II. His list price was \$150 and his show price was \$100. Computer Applications was selling][in a Mac at \$75 retail and \$50 show price. They didn't have a II there for demonstration purposes, but at only \$50 I took a chance.

You need a Super Serial card for your II and programs are transferred via either an Imagewriter cable or modem. A version of DOS 3.3 is supplied so that you can boot up your Mac into a II operating DOS 3.3 to run the transferred programs. A joystick, clock, and four disk drives are built into the screen. Both Integer and Applesoft Basic are also supplied.

I transferred Brickout and found it to run at about one-third speed. I then transferred my stock analysis program and found the calculation of a ten-day moving average for 200 days to be AGGONIZINGLY slow (15 min. vs 30 sec. on the II). I guess the utility of this program depends upon how much slowdown of the transferred program you can live with.

Tips on the LaserWriter and Other PostScript Printers and Typesetters

Copyright ©1986 Arthur Naiman. All Rights Reserved.

This article excerpted, with permission, from *The Macintosh Bible* by Dale Coleman and Arthur Naiman, to be published in February 1987.

Basic tips

Versions of software, models of hardware

Unless otherwise specified, these tips assume you're using the following software and hardware: Finder version 5.3, System 3.2, LaserWriter 3.1, LaserPrep 3.1, HFS, regular LaserWriter (not a LaserWriter Plus) and either a Mac Plus or an Enhanced 512K Mac (or any other machine using the 128K ROMs). If you're running with older versions of software than those, you should definitely install the new ones.

After installing a new printer driver

The first time you print from an application after installing a new version of a printer driver, choose Page Setup (on the File menu) before choosing Print. You don't have to change anything in it if you don't want to—you can just click OK as soon as the dialog box appears—but you do need to enter and exit Page Setup to activate the new driver.

Where to put the LaserWriter

The LaserWriter's fan makes a fair amount of noise. It's nothing like the one on a PC XT, which sounds like a commercial jet testing its engines, but next to the wonderfully silent Mac it can be pretty annoying. (This is particularly true if you're lucky enough to have a quiet place to work. In the typical office environment, you may barely be able to hear the LaserWriter).

One way around this problem is to put the LaserWriter in a closet. Since it's connected on AppleTalk, there's no problem with the cables not reaching, and most closets are large enough. According to the manual, you have to leave 7" between the rear of the LaserWriter and the wall, about 13" on each side and 16" to the front (where the toner cartridge is inserted). But you can cheat on the front dimension, because you can replace the toner cartridge with the closet door open.

We keep our LaserWriter in a closet with just 8" to the front, 7" to the rear and a couple of feet to each side, and we've had no problems with insufficient ventilation or heat buildup. Because closets usually have no windows, and often have no external walls, they tend to vary in temperature less than regular rooms.

If you're really worried about heat buildup, hang a thermometer on the closet wall and check it regularly. We do that, and ours has never registered above 80°—even on days when it was hotter than that outside and when the LaserWriter had been on for many hours (although, admittedly, it never gets *really* hot here in Berkeley; the average high temperature during the hottest month is just 72°).

The manual says the temperature of the air around the LaserWriter shouldn't get over 90°, which means that if you live in New Orleans or some other place where the sidewalks melt in the summer, you're going to need air conditioning to stay within their specs (which also call for humidity of 80% or less). But if you live in a place like that, the LaserWriter's not going to be the only reason you need an air conditioner.



There is, of course, one major disadvantage to putting the LaserWriter in a closet—you have to get up and walk over to it each time you want to look at the output. We don't find this bothersome, and enjoy the break and exercise (if walking ten feet can be called exercise). But it could get annoying, particularly if you're doing a lot of trial-and-error futzing with a document.

Flashing messages and lights during printing

When the LaserWriter and the Mac are trying to figure out how to print your document, a message appears on the Mac's screen that lists its "status" as "processing job." Every five seconds or so, this message flickers. DO NOT PANIC (as Arthur, of course, did). The periodic flashing doesn't mean that the power pole outside your house is about to fall over, or that enemy aliens from Saturn are trying to destroy your Mac by sending power surges through the house wiring. It's perfectly normal, just a way of reminding you that the Mac is thinking.

The LaserWriter lets you know it's thinking by single-flashing its yellow light every two seconds. (Double-flashing indicates a "wait state," which usually doesn't last more than 30 seconds).

When the Mac's role in the printing collaboration is done, the status message will disappear from the screen, but nothing may still have emerged from the LaserWriter. DO NOT PANIC (as Arthur, of course, did). The LaserWriter is still thinking on its own about how exactly to print the document (you know that because the yellow light is still flashing). You're free to continue editing the document, to close it, to quit the application, or whatever. The LaserWriter will print the document eventually, all in its own good time.

Printing multiple copies

Because the LaserWriter takes a long time to figure out a page, and not very long after that to print it, it's *much* faster to ask for multiple copies of a document than to reprint it several times.

Maximum image areas

The maximum area a LaserWriter will print on a standard US letter-size (8.5 x 11) sheet of paper is 8 by 10.92 inches, centered on the page. The width limitation is the most significant; it means you must always have a margin of at least a quarter inch on each side. (The required .04-inch border top and bottom is, of course, trivial).

What's more, few programs are capable of filling the entire area. *MacDraw*, for example, can only fill an area of 7.68 by 10.16 inches, and *MacWrite* requires a left margin of one inch.

The image-size restriction is even more dramatic on legal-size paper (8.5 x 14). There the LaserWriter can only fill an area 6.72 inches wide by 13 inches deep, thus requiring margins of more than seven-eighths of an inch on each side and half-inch borders top and bottom. (There are similar restrictions for the common European paper sizes, A4 and B5.)

The reason for these limitations is the LaserWriter's memory. Even though it has 1.5 megabytes of RAM, that's only enough to image between 87 and 88 square inches of page (which is what both 8" x 10.92" and 6.72" x 13" amount to). To really process a page adequately with PostScript, a printer should have at least 2 megs of memory, and ideally 2.5. Many other PostScript printers and typesetters do have that much memory but the LaserWriter scrapes by with 1.5 megs.



One time it's important to remember these size limitations is when you're proofing something on the ImageWriter that will ultimately be printed out on the LaserWriter. Because the ImageWriter can print wider than the LaserWriter, be sure to leave adequate margins; otherwise your image will get cropped on the edges when you put it on the LaserWriter.

How to get the right right margin

There's something very screwy about the right margin when you use *MacWrite* on the LaserWriter. To get a right margin of one inch, you think you'd set the right margin marker to 7-1/2" (8-1/2" -1"). But if you do that, you'll end up with almost no right margin at all. The setting that actually works is—for some strange reason—6-7/8". To get a right margin of 1-1/4", you set it at 6-5/8".

The rule seems to be: add 5/8" to whatever width you want the right margin to be, then subtract the total from 8-1/2" to get the correct setting.

How to get rich, deep blacks

You can't. Black areas of any significant size will contain small white splotches or streaks, regardless of where you put the *print density dial* (on the back of the machine). The Canon LBP-CX "marking engine"—around which the LaserWriter and many other inexpensive laser printers are built—simply can't do it.

This is only a problem with areas of black, not regular text—which tends to look good no matter how low you set the print density dial. And it's also only a problem if the LaserWriter's output is used at the final product. The film used at printer's will almost always fill the black areas in, and you can usually even get the same result from a photocopy machine.

But if you aren't going to print or photocopy what comes out of the LaserWriter, it makes sense to follow the rules below:

1. Try to avoid graphics with black backgrounds or other dark areas.
2. If, like most people, you'll mostly be printing text, use a page of text, not a graphic image, as your sample when you're adjusting the print density dial.
3. Don't bother turning the dial all the way to high—you'll just be wasting toner for no purpose.

Relative resolutions of various devices

In a normal ImageWriter text printout (from a word processing program like *MacWrite* or *Word*), there are 80 dots per inch across and 72 down; if you choose Tall Adjusted, there are 72 x 72. So the resolution of an ImageWriter printout varies between 5760 dots per square inch (in a regular Tall printout) and 5184 dpsi (in a Tall Adjusted printout).

The LaserWriter's resolution is 300 dpi, which amounts to 90,000 dpsi—about 16 to 17 times the ImageWriter's. But 90,000 dpsi is nothing compared to what regular typesetting machines can do. On the 1270-dpi Linotronic 100, for example, there are 1.6 million dpsi—about 18 times the resolution of a LaserWriter and almost 300 times the resolution of an ImageWriter. On the 2500-dpi Linotronic 300, there are 6.25 million dpsi—almost 70 times the resolution of a LaserWriter and well over 1000 times the resolution of an ImageWriter.



Fonts on the LaserWriter

ImageWriter fonts on the LaserWriter

If you turn Font Substitution on (in the Page Setup window), the LaserWriter converts four basic ImageWriter fonts to laser fonts when it runs across them in a document. The conversions are: New York to Times; Geneva to Helvetica; Monaco to Courier (they deserve each other); and Seattle to a modified version of Helvetica. But the character spacing produced with Font Substitution leaves a lot to be desired..

If you turn Font Substitution off, or use any ImageWriter fonts other than the four mentioned above, the LaserWriter creates special PostScript versions of the ImageWriter fonts it finds in a document. (Most people don't realize that PostScript can create *bit-mapped fonts* as well as *outline fonts*.)

These bit-mapped LaserWriter versions of ImageWriter fonts print more smoothly than the original fonts do on the ImageWriter, and they also scale *much* better. But rotation doesn't work too well (which means that they may not look great in italic, for example). Two other disadvantages of the PostScript versions: they take much longer to print, and they strip out most of the type style variations you get on the ImageWriter. (For more information, see the tips below titled "How fonts affect the LaserWriter's speed" and "Type styles on the LaserWriter.")

Two types of laser fonts

The LaserWriter comes with four fonts (or, to use the traditional term, four font families) built into its ROMs—Times, Helvetica, Symbol and Courier (if that can be called a font). If you buy a LaserWriter Plus, or the Plus upgrade, you get an additional six fonts in ROM: Palatino, Bookman, Zapf Chancery, Zapf Dingbats, Avant Garde and New Century Schoolbook. As of this writing (late 1986), that's all the fonts you can get in ROM.

If you want other laser fonts, you have to buy them on disk and *download* (send) them to the LaserWriter. *Downloadable* fonts have some advantages over fonts in ROM. For one thing, there's a much greater selection of them. For another, they're much easier for the publisher to enhance, upgrade or update. And they tend to be cheaper, with some selling for as little as \$30. But they do take longer to print, and they do use up some of the LaserWriter's precious RAM (see the tips on those subjects below).

Two ways to download fonts

Laser fonts that reside on disk rather than in ROM can be sent to the printer two ways. You can "manually" download them to the LaserWriter before printing, or you can have them downloaded automatically during printing (you just use the screen fonts in your document and the Mac sends the fonts' print files to the LaserWriter as it runs across them). Automatic downloading takes about 5-15 seconds per font.

(Apple and Adobe refer to manual downloading as "permanent" downloading, which is pretty confusing, since the fonts only stay in the LaserWriter until you turn it off. "Manual"—which is obviously based on an analogy to automatic and manual transmissions in cars— isn't exactly the right word either, but we can't think of a better one).

Manual downloading is done with a program that comes on the disk with the fonts, called Font Downloader, LW Download or some name like that. Automatic downloading only works with versions 3.1 or later of LaserPrep and LaserWriter (or with equivalent versions of *PageMaker's*



LaserPrep equivalent, AldusPrep) and with compatible system software. Most, but not all, applications support automatic downloading.

The advantage of manual downloading is that the LaserWriter prints faster. The advantage of automatic downloading is that it's easier (manual downloading is an annoying chore, even though it only takes a few minutes) and that the fonts get flushed out after each printing job, making room for new fonts on the next printing job.

As a rule, manual downloading isn't worth the trouble unless you're going to be using the same font(s) in three or more printouts (which might be the same document revised and reprinted three times). On the other hand, it almost always makes sense when you're printing from *PageMaker*, which flushes automatically downloaded fonts after each text block, rather than at the end of the printing job, and thus has to reload them for each subsequent text block.

Where to store downloadable fonts

If you're running under HFS, put the files for downloadable laser fonts either in the system folder (that's the best place for them) or in the startup disk's window (also known as the "root directory"). If they're stuck away in a folder anywhere else, the Mac won't be able to find them and they won't be downloaded to the LaserWriter.

Why those restrictions? Because there can be so many files on a hard disk that the Mac could spend forever looking for the fonts, if it didn't limit its search to just those locations. (However, the public domain program *Set Paths*, by Paul Snively, lets you put printer files wherever you want them.)

The above applies only to the actual printer font files that get sent to the LaserWriter; screen fonts should, of course, be installed in the System file with the Font/DA Mover (version 3.2 or later).

By the way, don't change the names of those print files; the automatic downloading process depends on their staying the same. (Sometimes the icon for a print file will change when you copy it from the disk you get from the publisher, but don't pay any attention to that).

Installing screen fonts

You can install no more than 200 ImageWriter fonts or screen fonts in the System file. You may think you'd never need to have anything like that many fonts installed, but each different size counts as a different font; in other words, it's the listing in the Font/DA Mover window (with the sizes alongside the names) that counts, not the one on the Font menu (where the sizes aren't listed). So if you have twenty fonts installed, each in 9-, 10-, 12-, 14-, 18- and 24- point, you're already up to 120.

Needless to say, you don't need to install every size of a font, unless you're going to be printing it out on the ImageWriter. Since the LaserWriter does its own scaling and doesn't depend on the screen fonts, you only need enough sizes installed to keep the screen display legible.

In our experience, performance begins to degrade significantly when you have anywhere near 100 screen fonts installed. Virtually anytime you try to install another font, Font/DA Mover (version 3.2) hits you with a bomb message ("ID = -2, -108"). You have to restart the Mac—which can take an annoying amount of time if you have a hard disk full of stuff—and then, having just sat through one restart, you have to Shut Down and sit through *another* one (this is the only



known way around the bomb message, and it doesn't always work). At that point, you *may* be able to install the font; if not, you'll have to go through a third restart.

This is a *very* frustrating bug which will hopefully have been fixed by the time you read this.

One way to keep the number of screen fonts down is to only install the ones for plain text, not bold or italic or whatever. If you have the 128K ROMs, choosing a plain text font and then making it bold (say) will produce exactly the same effect as if you choose the bold screen font. (For more on this, see the next tip.)

Type styles on the LaserWriter

On the ImageWriter, every font can be transformed into **bold**, *italic*, **outline**, **shadow** and any combination thereof—which comes to sixteen possible variations. On the LaserWriter, however, no font we've seen can produce all sixteen variations. For example, Zapf Chancery (from Adobe) only gives you plain, outline and shadow; bold, italic and bold italic print out as plain and all the other variations print out either as outline or shadow.

The LaserWriter's built-in fonts, Times and Helvetica, do much better. Although outline shadow looks the same as shadow, eliminating four possibilities, the other twelve variations are there. (We've yet to see a laser font that will make the distinction between shadow and outline shadow.) Adobe's Benguiat has the same twelve combinations, as does Casady's San Serif. Casady's Ritz provides eight of the possibilities—exactly half.

ImageWriter fonts lose most of their style variations when they're translated into PostScript bit-map fonts. For example, when printed on the LaserWriter, Chicago retains just five of the sixteen styles it has on the ImageWriter. The only way to tell what variations a particular font will give you on the LaserWriter is to try them out. (For an easy way to do that, see the description of font templates in the next section.)

All type styles on the ImageWriter are produced algorithmically—that is, by the application of a rule like 'increase the width 10%' (to create boldface) or 'slant right 15°' (to create italic). But on the LaserWriter, many fonts have their own separate *cuttings* for bold, italic and bold italic—that is, all the characters in those styles were individually designed, as if for a separate font.

If a font does have separate cuttings for certain styles, you'll have to load a separate printer font file in the system folder for each one. As a rule, you'll get better looking characters than with an algorithmically derived style.

You can always tell whether the style you're using is algorithmic by looking to see if there's a separate font file for it. (In some cases, you don't need to bother: for example, type styles called "oblique" rather than "italic" are almost always algorithmic.)

Fonts with separate cuttings usually come with screen fonts to match, but you can save room in your System file by not installing them. You can just select the type style you want from a style menu and PostScript will know to look for the font file before making the change algorithmically (at least it will on any machine that uses the 128K ROMs). But if you do install the special screen fonts, you'll get a cleaner representation of the characters on the screen.

Some fonts also have separate cuttings for variations with different names than bold, italic, outline or shadow (see the next tip for several examples). In these cases, you do have to install



the screen font—unless they style is basically a variation on bold or italic. If it is, it will normally be “mapped” to the menu option. For example, “demi” type is bolder than regular, so selecting text set in Souvenir and then choosing Bold from the menu will give you Souvenir Demi.

Generally, font publishers try to do what makes sense; if a font comes with four variations, you can usually get them by choosing plain text, bold, italic and bold italic.

Although “oblique” is the only other name in common use for italic type (both are slanted, but italic letters are also reformed in other ways), there are many names for various levels of boldness. The common ones are listed below, going from lightest to boldest:

Ultra Light
Extra Light
Light
Roman, or Book, or no adjective (just the name of the face)
Medium
Demi
Bold
Extra Bold, or Heavy, or Black
Ultra Bold

Special type styles on the LaserWriter

Some laser fonts offer exotic type styles far beyond the standard sixteen (only on the Mac could you call bold italic outline shadow “standard”). In addition to all the different levels of boldness mentioned at the end of the last tip, there’s condensed or “thin” (letters squeezed together stretched vertically) and expanded or “fat” (letters spread apart and stretched horizontally). Century’s Special Effects use some really fancy algorithms—stack, fill, reflect, etc.

(We’d show them to you here, but—given the software we’re using and the LaserWriter’s skimpy RAM—this document won’t print out if we put any more fonts in it. For an explanation of why that is, see the tips below on RAM limitations and using lots of fonts.)

To get special type styles like condensed and expanded, you have to install the screen fonts, since there are no menu items that correspond to them.

Creating your own special typographic effects

To distort a headline or other piece of display type for special effect, type it first in *MacDraw* or *MacDraft*. Then Paste it into *MacWrite*, *Word* or *PageMaker*, either through the Clipboard or the Scrapbook. Once it’s there, select it and drag on the little handles to stretch the type horizontally, vertically, or both. It will look terrible on the screen but when you print it out, the characters will have the same crisp, clean edges that laser fonts normally do.

You can also use *MacDraw* to “drop out” (“reverse out”) type—that is, to generate white type on a dark background. (But remember that black comes out streaky on the LaserWriter, regardless of where you set the print density dial, and you’ll have to photocopy or photograph the output to fill in the blacks.) To drop out type, first create a solid shape filled with black (or grey or any other pattern you want for the background). Then, with that shape selected, choose the font and size you want, choose Outline from the Style menu, and start typing.



You can also combine both these techniques, creating white writing on a dark background and stretching it once it gets into your word processing program.

These typographic special effects also work on the ImageWriter, but you don't get smoothing and the results don't look anywhere near as good. On the LaserWriter, however, you can really produce some nice-looking stuff, as the four examples below may or may not make obvious:

Gloria Zarit

Gloria Zarit

Gloria Z

Gloria Z

(The first two samples are in Adobe's Benguiat and the last two in Casady's Calligraphy.)

How fonts affect the LaserWriter's speed

The first basic rule is: the more fonts you use, the longer it will take the LaserWriter to print a page. (If you use too many, it won't print the page at all. For advice on what to do then, see the tip below about using lots of fonts.)

The LaserWriter processes some kinds of fonts much faster than others. Laser fonts in ROM are the fastest. Downloadable laser fonts are slightly slower (and, in addition, ones that are downloaded automatically take 5-15 seconds longer than ones that are manually downloaded, since that's about how long it takes to send them to the LaserWriter). ImageWriter fonts are far and away the slowest, because the LaserWriter not only has to download them but also has to create a PostScript version of them.

So you'll get the very fastest speed out of the LaserWriter if you use just one built-in laser font in just one style— and, because scaling also takes time, in just one size. But that's quite a sacrifice to make for a little speed.



RAM limitations

After the LaserWriter finishes imaging the page and doing other necessary tasks, it only has about 210K of RAM available for downloading fonts and other necessary information. To give you an idea of just how limiting that is, consider that *PageMaker* takes about 50K for its *header* (the PostScript instructions it gives to the printer) and that the necessary QuickDraw information takes another 90K. This leaves only 70K for your downloadable fonts, which usually require between 20K and 35K each.

(Fortunately, *PageMaker* flushes out the fonts after each text block, instead of at the end of the document, so the limitations are on fonts per text block rather than fonts per document. Other programs often have somewhat smaller headers, and thus leave slightly more room for fonts to be downloaded.)

In addition to the space the fonts themselves take up, another 10K or so has to be downloaded along with the first font. This *one-time hit* contains information that applies to all fonts.

In most applications, using the full image area only leaves you room to download three or four fonts; under some extreme circumstances, there may only be room for two. (Adobe's Font Downloader program has a menu item called Printer Font Directory that tells you how much RAM is left in the LaserWriter and what fonts—both downloadable and ROM-based—are already there. Casady and Century include similar programs on their font disks. All three companies also provide software for resetting the LaserWriter—that is, flushing its memory and making it generate a new startup page—without turning off the power).

To get around this limitation, some applications use *note format*, which gives half-inch margins all around (on the assumption that you're not going to want to print anything closer to the edge of the paper than that). This gives you room for more fonts, theoretically as many as eight or ten. But don't count on it.

By the time you read this, Apple will probably have released a new LaserWriter driver that will automatically swap fonts in and out of the LaserWriter's RAM *during* a printing job, rather than merely flushing out all the ones that have accumulated at the end of each printing job. (Some laser fonts may have to be rewritten to take advantage of this new feature.)

It's also possible, and devoutly to be hoped for, that future models of the LaserWriter will have more RAM available or, even better, that Apple or someone else will come out with a *font server*—a hard disk or other device that stores fonts and feeds them to the LaserWriter as needed (this probably would require laser fonts to be rewritten).

In the meantime, you'll almost certainly find yourself overflowing the LaserWriter's RAM. For what happens when you do that, see the next tip.

Using lots of fonts on the LaserWriter

Don't. (We'll resist the temptation to make this the shortest tip in the book.)

You can usually get away with three or four fonts—but remember that if bold, italic, bold italic or some other style comes in its own *cutting* (that is, if it's been separately designed and comes as a separate font file), it counts towards the total just like a completely different typeface. (One exception to this are special cuttings labeled "oblique." They're usually derived algorithmically—that is, by the application of a rule—and take up much less of the LaserWriter's memory. For example, ITC Glypha Oblique only takes up 3K if you already have



Glypha Roman loaded, since all that needs to be added are the PostScript instructions on how to tilt the letters.)

PageMaker offers you one way around the problem. With it, the limitations that normally apply to fonts per document apply to fonts per *text block*. And since you can have as many text blocks as you want in a document, you can also have as many fonts as you want (within reason, that is; things can get pretty tedious with dozens of text blocks on a page).

If you use too many fonts, the Mac will show you one or more messages that tell you you're running short of memory. In a classic case of boneheaded design, these messages neither beep nor stay on the screen for very long. So if you happen not to be completely catatonic and have therefore chosen to get up and do something rather than to stare blankly at the screen for the five or ten minutes it takes the LaserWriter to print a document with a lot of fonts, you'll miss the messages. When you come back into the room, the LaserWriter will be humming innocently away ("What do you mean, 'Where's your printout?' What printout?"), with no indication on the screen of what happened.

(The above assumes your fonts are being automatically downloaded. If you're manually downloading fonts—sending them to the LaserWriter one by one with a downloading program prior to printing—and you overrun the LaserWriter's RAM, it will probably just restart itself and spit out a new startup page.)

If a document won't print, it's always worth trying to print it again; it often works the second time. If that doesn't work, try printing the document one page at a time. If *that* doesn't work, try breaking it into smaller documents.

If you're not sure that too many fonts is the reason the LaserWriter won't print your document, the easiest way to find out is to make a copy of it, change the entire copy to Times or Helvetica, and try printing that. If that works, too many fonts is almost certainly your problem.

All in all, the limitation on how many fonts you can use is one of the LaserWriter's most annoying shortcomings.

Sticky header files

Most Macintosh applications send *header files* of PostScript instructions to the LaserWriter at the beginning of each document to be printed. Sometimes these header files stay in the LaserWriter and force it to reset (and spit out a startup page) before letting you print a document from a different application. This is a bug. If you find the LaserWriter resetting every time you switch from a given application, contact the publisher of that application for a fix.

Using downloadable fonts with 400K floppies

If you're using 400K disks, you may not have enough room on your startup disk for both the screen fonts and the font files you need in order to use downloadable laser fonts. (With ImageWriter fonts, you only need the equivalent of the screen fonts, so you can squeeze more of them on a disk.) The solution, of course, is to upgrade to 800K disks—which, in our opinion, is worth every penny of its relatively modest cost.

Copy protection on Adobe fonts

The only copy-protected laser fonts are those from Adobe Systems. When you install an Adobe font, you have to name the printer you want it to work with. From that point on it won't work



with any other printer—and, needless to say, you only get to install each font once. (Adobe does offer a more expensive version of their fonts that can be installed on up to five printers.)

Laser fonts in MacPaint

MacPaint, that bastion of bit-mapping, will not output laser fonts. What you get instead is the screen font, just the way you see it on the screen. To get around this, use *MacDraw* or *MacDraft* or, even better, *SuperPaint*, which lets you combine bit-mapped and object-oriented graphics.

Incorrect character spacing with the old 64K ROMs

When Apple introduced the new 128K ROMs (for the Mac Plus and the Enhanced, but also available as a separate upgrade), they allowed for a new font resource, *.FOND*. Unlike the earlier *.FONT* resource, *.FOND* is much smarter about character widths in laser fonts, and virtually eliminates the problem of spacing on the screen being different from what you get on paper (but see the next tip).

The 64K ROMs don't know about *.FOND* and therefore can't look for it. So if you want precise character spacing with laser fonts, make sure you've got the new ROMs.

Different versions of Times, Helvetica and Courier

The 10- and 12-point sizes of the laser fonts Times and Helvetica and the 10-point size of Courier have been slightly modified to improve character spacing. As a result, a document that was perfectly formatted in an old version of one of them may have some unexpected line breaks and page breaks when printed with a new version (or vice versa, of course). The only solution is to go through and readjust things as needed.

Images on the LaserWriter

Automatic reduction of some graphics

Some applications will scale bit-mapped images to 96% of their original size when printing them on the LaserWriter. That's because the Mac's resolution of 72 dpi doesn't divide evenly into the LaserWriter's resolution of 300 dpi. Reducing the image to 96% gives you 75 dpi, which goes into 300 dpi exactly four times. This helps with Smoothing.

(If the application doesn't do it for you, you can do it yourself, by simply putting 96% in the Reduce or Enlarge box in the Page Setup window.)

Halftoning on the LaserWriter

The LaserWriter handles text so nicely, it's tempting to use it to produce everything that goes on the page, including pictures. But certain kinds of images have to be processed before you can print them. Here's why:

All printed material is made up of either text or *art* (which is what every graphic element except text is called). There are two kinds of art—*line*, which contains no grays, and *continuous-tone*, which does (the name comes from the fact that the tones form a continuum, from black through gray to white).

Continuous-tone art present a problem: Since printing ink is black (or whatever) and paper is white (or whatever), how can the grays be represented?

Far-and-away the most common solution involves putting a *screen* over the photograph. (It's called a screen because the original ones were made of fine metal mesh, although most screens



today are sheets of plastic with dots printed on them.) The screen converts light grays to tiny black dots on a white background, and dark grays to tiny white dots on a black background. From a normal reading distance, these dots—in many different sizes—look like various shades of gray.

Once a continuous-tone image has been screened, it's called a *halftone*. If you hold a magnifying glass to a halftoned picture (virtually any printed picture will do), you'll see the little halftone dots, but you won't find them on a continuous-tone photograph like a snapshot.

If you use the traditional screen approach to halftoning, you won't be able to integrate your graphics with your text and print them both out on the LaserWriter at the same time; you'll have to leave a space for the graphics and add them at a later stage. To integrate halftones with text, you need a device called a *scanner* that can capture continuous-tone images and halftone them electronically.

Scanners vary in *sampling density* (how many times per inch they evaluate the picture) and in how many levels of gray they'll pick up. They also vary in quality and price. ThunderScan is an inexpensive scanner (\$230) that can produce images suitable for use in an informal newsletter—or better than that, if you really know how to use it. Microtek and Abaton, which scan at 300 dpi, cost about ten times as much; their software makes it easy to capture an image as a PostScript file. An even higher-quality scanner is the ImagiTex 1085.

Once the scanner has captured the image, you can use various programs—usually software that comes with the scanner—to play with it. But you might be better off simply capturing the image as PostScript data and letting the LaserWriter do the halftoning—if the scanner will let you do that. (For one way to turn *MacPaint*, ThunderScan and MacVision documents into PostScript files, see the next tip.)

PostScript has better halftoning algorithms than most scanners, and also has many commands for altering the image. (The second issue of *Colophon*—the newsletter of Adobe Systems, which developed PostScript—shows several samples of what it can do.)

The quality of a halftone depends on the *frequency* of the screen you use (how many lines per inch it breaks the picture into). The higher the number of lines, the better the photograph looks. The coarsest screen in general use is 65 lines per inch. Newspapers typically use an 85-line screen and magazines a 120-line screen. The finest screen in general use is 150 lines per inch.

Laser printers and typesetters each have a *default screen frequency* that the manufacturer has picked to produce the best output on that particular device. On the LaserWriter, it's 60 lines per inch (*lpi*); on the Linotronic 100, 90 lpi; on the Linotronic 300, 120 lpi. You can vary this frequency somewhat (using simple PostScript programming), but if you go too far from the default, the results may not look very good.

The LaserWriter's resolution of 60 lpi is just below the usual minimum standard for halftones, and you can't expect an image printed at that resolution to look like the ones in magazines and books. But the quality is good enough for many purposes.

Converting images to PostScript files using LaserTools

JustText is a powerful but complex text formatting and page makeup program that includes a useful set of utilities called LaserTools. One of these, Paint to PostScript, converts MacPaint



files to PostScript image files, which can then be manipulated in many ways that aren't possible with MacPaint. LaserTools also includes utilities for converting ThunderScan and MacVision documents to PostScript.

Supplies for the LaserWriter

What kind of paper to use

Laser printers put images down on paper in exactly the same way as photocopiers do; in fact, the guts of a LaserWriter (and of several other PostScript-compatible laser printers) are identical to those of many Canon copiers (all use the LBP-CX "marking engine"). So the kind of paper specifically designed for use in copiers—often labelled "xerographic"—is what you want for the LaserWriter.

Copier paper comes in more than one grade. When you're preparing originals for presentation, you'll naturally want to use a nice-looking, heavy, opaque paper. But for everyday use, and when you're preparing documents to be reproduced, use the cheapest kind you can find. Here's why:

When the humidity is high, pieces of paper tend to stick together. To help them separate more easily, paper manufacturers put powder between the sheets. This is called *dusting*, and the more expensive a paper is, the more dusting it tends to have. The problem with dusting is that particles of the powder tend to get bonded to the paper along with the image, producing a rough, uneven surface. So cheap paper, with little or no dusting, is best.

Cheap paper has another advantage: because of its low fiber content, it has a smoother surface than most expensive paper (except for some types specifically designed to be ultrasmooth). The smoother the surface, the more precise will be the image bonded to it.

If there's a discount paper supply house in your area that sells retail, you should be able to buy inexpensive copier paper for \$2-\$3 a ream (500 sheets). If there isn't (and you live on the West Coast), try a local Copy-Mat outlet. They sell the standard paper they use in their machines for \$3.75 a ream. (We don't know if other copy shops also sell their paper, but it's worth asking.) Copy-Mats also have a fairly good selection of other kinds of paper (colored, "parchment," "laid," etc.), although a paper supply house's stock will obviously be much more extensive.

Except in an emergency, it makes absolutely no sense to go into a stationery store and pay \$6-\$7 a ream when you can get the same paper at half the price or less elsewhere.

Breaking in new toner cartridges

Before installing a new toner cartridge in a LaserWriter, you should rotate it gently from side to side (that is, lift the right side, then the left—it's roll you want, not yaw). If you don't do this, the images you get may not be dark enough. (You may also have to break in the cartridge by printing about thirty pages.)

The LaserWriter manual has an excellent section (chapter 4) on all aspects of setting up and maintaining a LaserWriter; it's clearly written and full of helpful illustrations.

Buying new toner cartridges

If you can't find an Apple-brand toner cartridge for your LaserWriter, the Hewlett-Packard LaserJet cartridge will work just as well. The cost to the dealer for both brands is the same, so your cost should be too. In theory, a toner cartridge made for any other laser printer that uses the Canon LBP-CX marking engine should also work, but we haven't tried them.



Miscellaneous tips

Proofing on the LaserWriter before typesetting

The Macintosh and the ImageWriter use QuickDraw routines to create their images, while the LaserWriter and many other laser printers and typesetters use PostScript. Although a lot of brilliant work has been done to allow QuickDraw and PostScript to talk to each other, what you see on the Mac and an ImageWriter is always going to vary somewhat from the output of a PostScript-driven device.

This means there's no substitute for hard copies. If you're planning to do final output on a PostScript-driven typesetter like the Linotronic, you definitely should proof your work on the LaserWriter. The differences between its output and the typesetter's will be negligible. (The resolution will, of course be higher on the Linotronic—that's why you're using in the first place—but the position of all the elements, and the overall look of the page, will be the same.)

There are two exceptions to that rule. First of all, the Linotronic can print all the way to the edges of the roll of paper it's using (which can be either 8.5" or 11" wide) while the LaserWriter can't print wider than 8". Secondly, the Linotronic won't smooth bit-mapped images. If you want the images smoothed, print them out on a LaserWriter and then paste them manually over the equivalent unsmoothed images in the Linotronic output.

Renting time on a LaserWriter (or other PostScript device)

Businesses are springing up all over (particularly in big cities and near universities) that let you come in with your Mac disk and print it out on one or more kinds of PostScript-compatible printers. Renting time can be surprisingly inexpensive; there's a place near us that charges just \$5 an hour and 15¢ a page.

If you can't find a place in the phone book, check with a local Mac user group (in fact, do that first; unlike the Yellow Pages, they can tell you which businesses know what they're doing, charge the least, and so on). If there's no user group in your area, or they don't know of a rental place, check out the tip below about remote typesetting services; many of those services also provide LaserWriter output.

Getting your disks ready for a rented LaserWriter

When you're planning to rent time on a LaserWriter, you might as well format your documents and set up your disks as completely as possible ahead of time, to cut down on the amount of time you have to pay for.

The first step is to make sure that the disk you bring with you has the software you need on it; the rental place will obviously have some software, but they may not have what you used to create your document (or the same versions of it). Call before to make sure they have the laser fonts you want to use installed on their printer; if not, be sure that the System file on the startup disk you bring has the appropriate screen fonts installed, and that the System folder contains the font files you need for the printer.

If you haven't already done so, change the fonts in your document to laser fonts by selecting the text and choosing the screen fonts just the way you would any other font. (We're assuming you proofed your document on the ImageWriter. Next, open Chooser (on the Apple menu) and select the LaserWriter icon. If you're using Chooser version 2.3 or later and System version 3.2 or later, you can do this without having a LaserWriter actually attached to your system—as long as you



have the LaserWriter file in your System folder and have connected AppleTalk on the Control Panel.

(If you're using older system software and have Choose Printer instead of Chooser, you can select 'no printer' by clicking in the space just below the last printer listed. But why are you using out-of-date system software? The new stuff is free.)

Once you've selected the LaserWriter in the Chooser, go back into your application, open the Page Setup window (on the File menu), make any changes you want (or no changes) and click on OK. Now go through your document. ImageWriter page setup has different margins from LaserWriter page setup, and there are other incompatibilities, so there may be changes you need to make. When you've done that, you're ready for the shortest possible time rental on the LaserWriter.

Remote typesetting from Mac disks

Let's say you've proofed your document on a LaserWriter until it's just the way you want it. Now you want to typeset it. But your budget's a little tight and you can't afford to buy a Linotronic this month (it probably wouldn't fit in the closet anyway). Don't despair. There are several services that will accept your Mac disks and print out from them on their own PostScript-driven typesetters or laser printers.

The costs usually run from \$4 to \$15 a page (some places also have a one-time registration charge of about \$50). Many of these services let you send data via modem—although if your document is at all long, it's going to take forever. Another advantage of mailing the disk (or, if you're in a hurry, sending it by Federal Express) is that you can send a printout along with it as a proof, so there won't be any questions about how you want the finished document to look.

Even if you send the document on the phone, you should send a message with it detailing exactly what you want. This is always a good idea, but with version 1.9 and earlier of *MacDraw*, it's an absolute necessity. *MacDraw* remembers fonts in the order in which they were installed in the system (other applications may do this too), and since the odds that the typesetting service will have installed the same fonts in the same order that you did are about one in a trillion, you'll need to tell them what fonts you want.

(This is not a problem with *MacDraw* drawings after they've been pasted into documents created by other applications—as long as the same System was used to create those documents as was used to create the *MacDraw* drawings.)

For more current information, check out the various magazines on the Mac and on desktop publishing; they sometimes run updated lists of such services.

Suppressing the startup page

Even time you turn on the LaserWriter, it spits out a rather attractive test page that tells you how many copies have ever been printed on the machine. It's possible to get into this constantly mounting total as a measure of your productivity, and therefore your general worth as a human being ("I've printed 3000 pages on my LaserWriter; I must be doing something useful with my life"), but it does cost you about 3¢ in toner, some fraction of a cent in paper, and some incomputable but probably significant amount in wear and tear on the machine.



So it's sometimes nice to be able to turn off the startup page, at least for a while. Century Software has a pair of programs that do that. Called Start-up On and Start-Up Off, they...well, we're sure you can figure it out.

Here are two public domain programs that do the same thing. Just type them out as you see them below (only the last line is essential), save them as a text only file with a name that's easy to remember, and send them to the LaserWriter with a program like PostScript Dump, Font Downloader or LW Download (which come on laser font disks).

```
%Disable LaserWriter from Printing Startup Page upon Powerup
% John Monaco Compuserve 73317,3677  %%GEnie J Monaco  %%EndComments
serverdict begin 0 exitserver statusdict begin false setdostartuppage
```

```
%Enable LaserWriter to Print Startup Page upon Powerup
% John Monaco Compuserve 73317,3677  %%GEnie J Monaco  %%EndComments
serverdict begin 0 exitserver statusdict begin true setdostartuppage
```

One function of the startup page is to tell you exactly when the LaserWriter is ready to begin printing. If you turn off the startup page, you need to know the following:

When you turn the LaserWriter on, it tests itself; while it's doing that, the green light blinks. When the green light stops blinking, wait a few seconds and the machine is ready to use. In a cold room, this warmup and test procedure may take about a minute. If you've used the machine recently and/or the room is warm, it might take much less time than that.

Printing custom stationery on the LaserWriter

Access to a LaserWriter can not only save you the expense of buying stationery, it can allow you to modify your stationery as often you like. You can change the text, the fonts, the graphics, or the paper you print it on. On the LaserWriter, it's just as economical to produce five sheets of letterhead with matching envelopes as it is to produce five hundred.

There's no special trick to doing letterhead on the LaserWriter. But envelopes can be tricky. You have to hand-feed them into the LaserWriter, face up and against the back edge of the manual feed guide (where the small brown label is). Certain kinds of envelopes don't feed very well—like ones made from “parchment”-type paper.

If you print your name and return address along the length of the envelope (that is, in the same direction as the name and address of the person you're sending it to), you'll only have to run the envelope through the Laserwriter once. If you want your name and address to run across the end of the envelope—which looks a lot snazzier—you'll either have run the envelopes through a second time, print labels and stick them on, or address the envelopes by hand.

Whichever direction you use for your address, you may find yourself frustrated by the LaserWriter's inability to print closer than a quarter of an inch to the edge of the envelope.

Word on the LaserWriter

Although Microsoft has given more consideration than most software publishers to optimizing their software for output on laser printers, they've gone about it in a nonstandard way. As a result, the performance of *Word 1.05* on the LaserWriter is quirky when running under System version 3.1 and later (and you certainly don't want to use an earlier version of the System).



BMUG

FALL 1986

This may also be the reason why some people say *MacWrite* prints out faster than *Word* on the LaserWriter.

There's no fix for this, but *Word 3.0* deals with the problem.

Viewing PostScript code

To see the actual PostScript code that gets sent to the LaserWriter, give the Print command, click OK when the print dialog box comes up, then immediately hold down COMMAND-F. (When you see the message that reads "creating PostScript file," you can let up on the keys.) The PostScript code generated to print the document will be placed in a file on disk, rather than being sent to the printer.

This file will be named "PostScript" and it's a good idea to rename immediately, because if you use COMMAND-F, a new file named "PostScript" will overwrite the old.



pen/Wood



Linotronic L100 Tips

Ron Schmidt

- Never use spaces to line up columns of text. Even if it lines up on the LaserWriter it won't on the Linotronic.
- Light grey shade fill patterns in MacDraw, PageMaker and other applications appear very light on L100 output compared to the LaserWriter output. In fact, the lightest fill patterns disappear.
- If you have installed and used *Italic*, **Bold**, or **Bold Italic**, screen fonts in one or more of your documents, make note of it when you submit your job for imagesetting.
- Some Linotronic systems only carries PLAIN screen fonts; Palatino, Symbol and Times. If your Font Menu carries other fonts that resemble B Palatino Bold, BI Palatino Bolditalic, I Palatino Italic ask for assistance because this can cause printing problems.
- Times, Helvetica, Symbol and Courier are ROM resident fonts. All other fonts are non-ROM resident fonts, and the styles Plain, *Italic*, **Bold**, and **BoldItalic** each count as one "downloadable font."

PageMaker will allow a maximum of 3 downloadable fonts per text block (i.e., text between two "windowshades.") There is no limit to the number of blocks per page. All other applications (MacDraw, Microsoft Word, etc) allow a maximum of 3 downloadable fonts per page.

Particularly complex blocks of text or pages, composed of text using certain memory-intensive downloadable fonts (i.e. Palatino, **Bold Italic**), may lower this maximum to 2.

- Documents which exceed the Linotronic's memory will require editing.
- MacDraw has a reputation for switching font names around. Make sure to submit a copy of the LaserWriter output with the font names written next to each text object.
- PageMaker can print a full 11 x 17 inch page on one continuous piece of film.
- Text prints slightly longer on the Linotronic than on the LaserWriter

Using the Hewlett Packard THINKJET and QUIETJET+ Printers

Kenneth William Hirsch

The June, 1985 issue of *Consumer Reports* gives a very favorable review of the Hewlett-Packard Thinkjet printer as used with IBM and similar computers. For six months, I have been using one with my Mac+. The printer requires a regular Imagewriter cable and a \$45 list (\$10-25 via mail order) program, *JetStart*.

Characteristics Of The Thinkjet And Quietjet+

What distinguishes the Thinkjet (and its new sibling, the Quietjet+) from Apple printers are their small size, quiet operation and, in comparison with the Imagewriter, superior printed image quality. The superiority is produced by the precision of the ink jet dots, which are very sharply defined. The Thinkjet, which lists for \$495, can be obtained from mail order discounters for about \$350. The Quietjet+, a newer, enhanced 15" platen version of the Thinkjet, lists for \$795. Both use the same clever ink reservoir + image formation print head. Both contain built-in fonts used in fast draft printing whose image quality I consider to be superior to that of the draft printing produced by the Imagewriter. Examples of two of the internal fonts are shown below:

(The Thinkjet/Quietjet + uses a replaceable cartridge, about the size of a spool of thread, to supply ink and form character and graphic images. Loss of image formation accuracy from print head wear, a problem with dot matrix printers such as the Imagewriter, is prevented because each new ink cartridge contains a fresh set of metal ink flow slits which form the image. Cartridges cost \$8 and last about as long as an Imagewriter ribbon.)

Proportional-Bold Text Sample, as printed by the HP Thinkjet

(The Thinkjet/Quietjet + uses a replaceable cartridge, about the size of a spool of thread, to supply ink and form character and graphic images. Loss of image formation accuracy from print head wear, a problem with dot matrix printers such as the Imagewriter, is prevented because each new ink cartridge contains a fresh set of metal ink flow slits which form the image. Cartridges cost \$8 and last about as long as an Imagewriter ribbon.)

Compressed-Bold Text Sample, as printed by the HP Thinkjet

The Thinkjet (but not the Quietjet+) reduces the expected page image size by 24%. It centers the smaller outcome on the normal page, thus resulting in larger margins. The Thinkjet's image shrinkage usually improves the clarity of letters and numbers, for many type fonts look and read better when reduced. With image quality already enhanced by dot sharpness, this reduction improves the appearance of drawings and graphics.

Speed

The Thinkjet printer is reasonably fast when using its internal fonts in "text" printing mode. In Standard mode, the Thinkjet and Imagewriter I are 28% slower than my Diablo 630 (formed character daisy wheel) printer. (Xerox claims the Diablo prints 40 characters per second.) In



High Quality mode, the Thinkjet and Imagewriter I are considerably slower than the Diablo. Here are comparisons of time required to print this article, using identical type fonts: (Thinkjet and Imagewriter I = Dover 10 point; Diablo = Prestige Elite 10 point). As one can see, actual dot matrix print speed is often much slower than the performance claimed by manufacturers.

<u>PRINT MODE</u>	<u>IMAGEWRITER I</u>	<u>THINKJET</u>	<u>DIABLO 630</u>
Draft	142 seconds	182 seconds	---
Standard	328 seconds	326 seconds	234 seconds
High	496 seconds	720 seconds	---

The Joy Of Six

My usual routine before obtaining the Thinkjet was to get up very early to finish and print work which was abandoned in the wee hours of night-before-deadline. The problem with getting up and printing at six AM is that the rest of the household is not pleased at that time to listen to printer clack/whine/ ratchet. One solution would be to: seize about double the area of desk space required for a printer and allocate it to a sound reduction enclosure into which to exile the Imagewriter (or other dot matrix printer); stalk and capture the \$150--\$200 required for such a quality sound enclosure from funds allocated for other purposes; fumble through clumsy latches in the newly purchased sound enclosure to get at the printer to change ink cartridges and add paper, and get on with life. The Thinkjet/Quietjet+ have another solution. They are quiet. For me, this has been the great blessing in using the Thinkjet. Speed and small size are added benefits.

Libraries And Classrooms

The quietness of the Thinkjet/Quietjet+ makes their use in interactive training and demonstrations and in libraries (for on-line printout of bibliographic retrieval) quite feasible. Printing can go on while attendees continue to listen to the speaker. This means a very portable system, consisting of a Mac, a Thinkjet, and a Koala KAT (or other hand-held cursor positioner), can be set up on a cart and connected to either a projector or, via a composite adapter, to a large monitor or TV set. The work demonstrated may then be captured on a "hard copy" for the participants.

Problems

The printers have some drawbacks, including their need for special, more expensive paper, occasional image degradation from clogging of a jet, incomplete compatibility with some Mac programs and, with the Thinkjet, the inherent shrinkage of printed image.

Clogging

Occasionally, paper dust clogs one or more of the ink slits, causing image degradation. The fix is to remove the ink cartridge and poke it with the epitome of high technology, a straightened paper clip. It usually works. If one makes sure not to print across page-end perforations, clogging can be almost entirely eliminated. This seems to be about the only problem normally encountered; I have had no other machine problem, no mechanical adjustments to make, and no electronic failures in more than six months use.

Program Incompatibility

Softstyle, the Hawaii-based firm which writes *JetStart*, the program which enables the Mac to use the Thinkjet and Quietjet+, has been hard-pressed to keep up with Apple's changes in the Mac operating system. At present, the Thinkjet/Quietjet+ doesn't properly print *Overvue* documents (my version 2.0b), nor does it correctly print the Thinkjet's wonderful internal



fonts in Text mode when running *Catalog Keeper*. There may be other programs which are not yet compatible. I have used it successfully with *MacWrite*, *MacPaint*, *Word*, and *MacProject*. Softstyle is working on compatibility. However, it seems to take them a great deal of time to make needed fixes.

Thinkjet Image Reduction

Image shrinkage occurs on the Thinkjet because the printer takes one inch at left and right margins, uses that space for paper pin-feed wheels and rollers, and squeezes the material that has been fed to it from a document *including the space allotted for margins* into the 6 1/2 inches of paper width which remains. This results in about a 24% reduction of size of text and graphic images. We don't get the benefit of the shrinkage back in the form of extra lines and words on a page; we simply get larger top and bottom margins and the same number of lines on the page as are in the source document.

Thus, to make the compression/shrinkage of the Thinkjet into a virtue requires creating page formats which are longer than standard. Both *MacWrite* and *Word* have very limited capability to print more lines of text on a page. They allow adjusting top and bottom margins within the already established page lengths in the printing resources. Within the limitations of the normal 11 inch long U.S.A. page and the slightly longer European page, extra text lines can be shoved into the Thinkjet pages and still leave generous top and bottom margins. But I know of no way to place text nor graphics out to the top and bottom edges, nor to use longer paper such as legal size (8 1/2 inch x 14 inch) to obtain more text lines and longer graphics images. It may be possible to accomplish these objectives; I have not gone beyond experiments which cram a few extra text lines into top and bottom margins.

NOTE: Softstyle is working on an update of JetStart which will get back the space at the top and bottom of the page. The image size reduction/compression problem may disappear when Hewlett Packard brings out an 8 1/2 inch platen version of the Quietjet+ to (replace?) the current model Thinkjet. Rumor indicates they will do so about the end of 1986. Proces may go up, however, for the Quietjet+ version of the printer is an improvement--smaller, lighter, easier to insert and remove paper, and capable of using colored inks.

About the Author

Kenneth William Hirsch was recently confused with being a medical doctor, which he is not. (See "Word Handler".) He is a Professor of Communication Studies at California State University, Sacramento, where he is currently convening a new Sacramento University Macintosh group (SUMAC?), after having done much the same thing to help start the Sacramento Osborne Group (SOG) more than four years ago. SOG is still going strong. Hirsch makes no such claim for himself.

BMUG
Fall 1986



Hardware

HARDWARE

Speeding up the Old Mac 512 • When the watch seems to stop	Joseph Schaller	239
Why Everyone Should Get the New ROMs • The decision to upgrade	Gordon Haig	246
I'm Upgrading! • More reasons to beef up the Mac	Ken Wait	248
The heart of the Apple II is still beating • About the Apple IIGS	Linda Custer	249
Radio Shack Model 100 Portable Computer • Making it a Macintosh add-on	Gordon Haig	253
A Day at the Races with MacCharlie • In the IBM den	Vicken Khachadourian	263
Keyboard Wrist Rest • A do it yourself project for tired typists	Kenneth William Hirsch	266
Locking your Macintosh • Alternate security kits	Cliff Stoll	268
Hush, Micah! • Installing an on-off switch for a hard drive	Michael Lamoureux	269
Tips for Hard Disk Users • Getting the most out of your hard drive	Dan Callaway	271
Building Your Own Hard Disk • Assembling a SCSI disk from scratch	Tim Standing	274
Making Printed Circuit Boards on a Macintosh • Using MacPaint to design and make PC boards	Bryan McDonald	288
SCSI Hacking • General purpose I/O with the SCSI port	Michael Lamoureux	294
Adjusting Your Mac • With a voltmeter and guts, you can...	David W. Berry	300



Speeding Up the Old Mac 512, or...What to do Instead of Watching the Watchhands Spin

Joseph Schaller

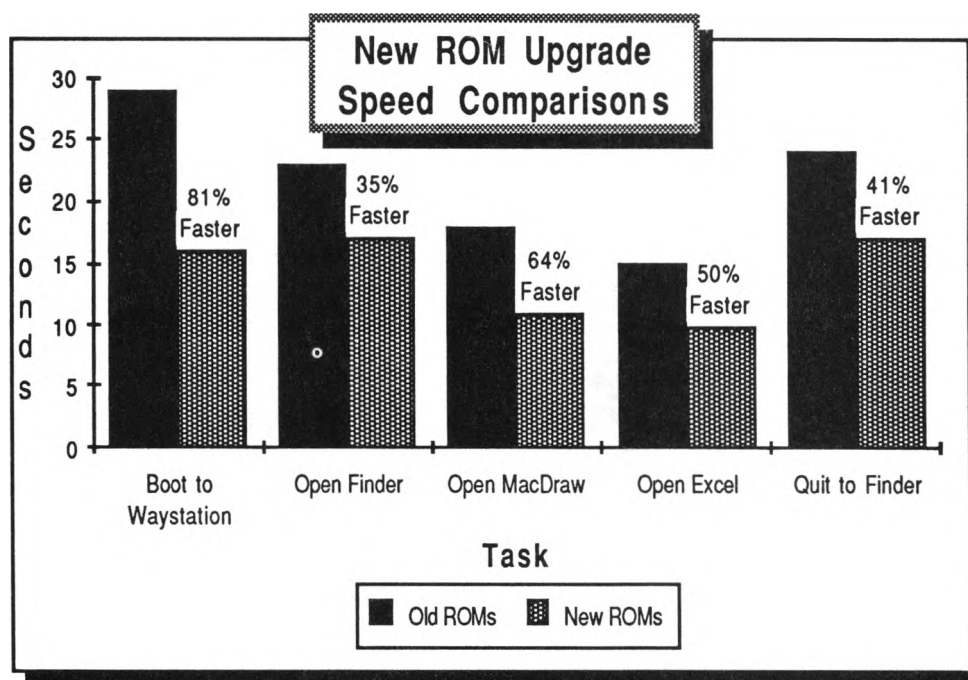
I can remember when I used to work on UCB's Unix system and I would get ticked if I had to wait five minutes for a response during busy periods. Now I get bored waiting 20 seconds for an application like *MacDraw* to open. Here's what I've gathered from my own experiments and from reading various sources:

New ROMs & 800K Upgrade

I could kick myself for not buying the new ROMs sooner. If you're a BMUG member, you can get them with the 800K drive upgrade for \$239 installed from Computer Selection in the City, \$300 list. But it's like GM offering to replace your year old 4-cylinder with a 6 and change the body to a station wagon for about 10% of purchase price. (Those of you still having tantrums about Apple's reasonable upgrade policy, take note: That works out to about the difference in list between the 512 & Plus.) And prices keeps dropping with volume & time.

The 800K drive is not only double the space, but about 25% faster and quieter. The ROMs alone, tested on the same exact configuration (in 5.3/3.2) make a 512K upgraded between 30% to 80% faster in getting between applications and the Finder. Any concerns you may have had about bugs appear to be alleviated by now. I've had less problems with new ROMs than old.

ROM Upgrade Chart



Tested on 512K Macs with MacBottom 20M HD using identical System 5.3 / Finder 3.2 by M. Joseph Schaller

System Fragmentation

Any system (Macintosh and MS-DOS included) using a 'where you can find space' approach to file management, instead of always-contiguous sectors, ends up having odd spaces here and



there as files are built, removed and changed. I've tested this on a fast PC, and a virgin copy of *dBase III+* will open in 6 seconds, a second one copied on after 2 weeks of development, in 10 seconds, and a third, added to disk after hundreds of editing changes on dozens of files (very fragmented) in 13 seconds. I've observed speedups similar in size after cleanup on the Mac; that is, a fragmented disk that used to take 40 seconds to boot, after backup/restore took 23 seconds. (With the new ROMs, 9 seconds.) The reason: Say File A is in sectors 1-10, File B in 11-20. You now append to A, which goes in sector 21, to B in 22, A in 23, etc. The work involved in directory lookup, reading one block (1-10), moving to the next block (21), settling down, reading that one, and so on, is responsible for the slowdown. Besides, it's more likely that your sectors get spread out by hundreds, not a couple. These factors are especially important on a hard disk.

Fix 1: 'backup/erase/restore' or a 'copy/erase/copy back'

File by file copying (by either backup or copy) will result in contiguous placement of sectors for each file on the backup disk if it's empty. Diskcopy or image backup obviously do not help; they result in the same random arrangement as before. 'Copy' is easier if you only have floppy drives than if you have a hard disk, because of sheer volume. You're talking about several hours of attended work, shuffling floppies.

You might have some impact if you just remount your System files alone, but then your other files may be still fragmented, and I don't know yet whether the System still gets spread around in the existing gaps or not. (Somebody want to test that?)

For hard disks: Tape drives are OK. A second hard drive (yours or somebody's new one...) could be cheaper, easier and more reliable. For floppy backup...get a good book and a big pot of coffee.

Fix 2: A Disk File Compactor

The real advantage of a Compactor is this: You can start it, go away, come back when it's done. No disk swapping, no restarting because of disk error, no getting them out of order...and when you're done, your Finder opens in, say, 9 seconds instead of 23.

On the PC I use the *Mace* utilities (\$79). On the Mac, *DiskExpress* from ALSoft (\$29 list). Both come with explicit warnings: Use them only on data you have safely backed up. So what's the advantage, then? Easy. You only have to do the complete backup once (and you should already have that anyway; however, most people 'get religion' about backups only AFTER they lose three months worth of work and data.) From then on, you only have to back up incrementally, just the files that have changed, which any intelligent backup program can do. We're talking \$19 discounted and a couple of floppies.

How do they work? *Mace* works well. But that's on the PC. Personally, I have had no trouble with *DiskExpress* on the Mac. But then, I have the Apple HD20 hard drive. I've also used it once on a client's Dataframe with no problems. Someone on MacQueue said there were problems with dynamic-volume-sizing drives like Hyperdrive, which doesn't surprise me; supposedly a fix is in the works. Again, only try it if you are behaving smart, i.e., have a backup of your data. But you'll like it much better than backup/format/restore.

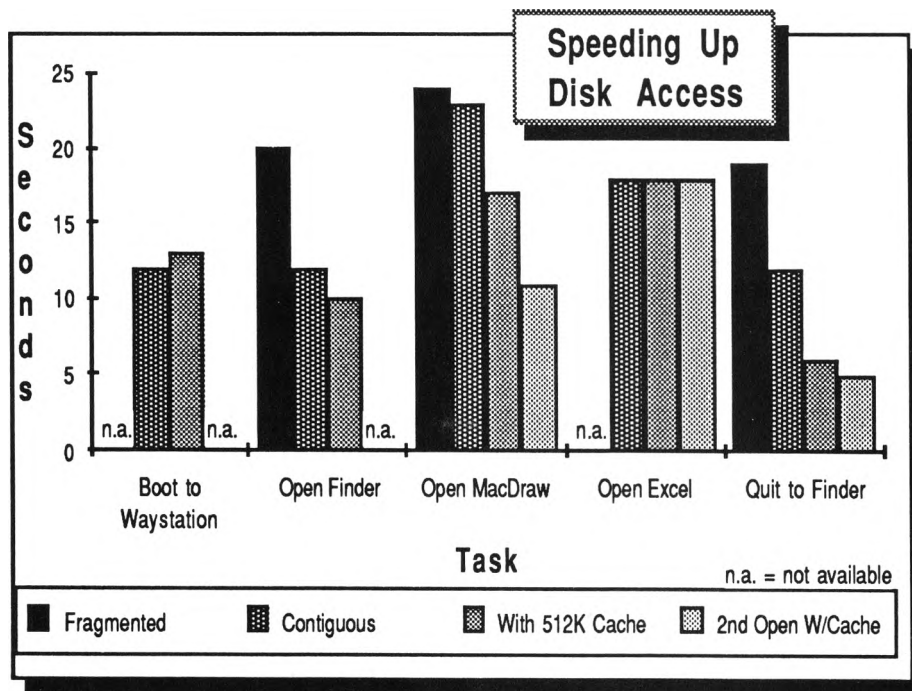
MacExpress and *Mace* work similarly: They take a catalog, examine for non-contiguous sectors, and then swap sectors, one pair at a time, until everything's in contiguous blocks. (That, as you can imagine, on 20 Meg, can take a while, like a couple of hours.) You can also specify in both an additional moving of all 'program' files to the beginning and data behind, since programs don't change size (as much). Both will do an 'analysis' first, to tell you how much needs doing. On the Mac, you can also specify compacting the Desktop file (about which more



later). *MacExpress* will not work on a disk that has any weird copy protection scheme installed (you manufacturers listening?).

What if it blows up in the middle? Right. Well, first of all: Don't. *DiskExpress* warns you not to, and tries to prevent it. The worst case should be that you lose one of the sectors being swapped, assuming they swap cautiously and keep the catalog up to date (Right, AI?). At very worst, you restore from backup.

Disk Access Speed Chart



Tested on an Apple HD20 with System 3.2/Finder 5.3, new ROMs by M. Joseph Schaller

Big Systems

The more you have in there, the more has to be loaded each time you return from or enter an application. You can get around this by not returning to the Finder, but still, you have to load fonts, build the Apple menu on the fly, in applications as well, even if you are using a launcher or minifinder.

Use *Font/DA Mover*.

You will find it helpful to remove:

- Fonts you never use
- Sizes of fonts you never use (except for doubles)
- DA's you rarely use (alarm clock, puzzle, control panel?)

And you can:

- Install fonts for graphics directly in *MacPaint* or *MacDraw*
- Use Other DA or *MacSampler* to temporarily mount DA's

Whatever you do, keep in mind that it is the installation and changing of fonts and DA's in particular that expands and fragments the System file. If you never do anything to it, it'll stay fine and fragmentation will only be a problem for your other files.



By the way, a bug was reported in one of the Systems (first release of System 3.2) in which a double copy of system resources was left in and took up twice as much room. Make sure you have the June 2, 1986 System.

Desktop

The Desktop, which keeps track of your files, grows as you add more files. I've read that it does not shrink when you delete files, or move them to another disk, or around in folders. Thus at one point, I had managed to increase my desktop from its normal 78K (I have over 900 files on my hard disk) to about 300K. This slows things down considerably. You can compact this Desktop file by holding down the option and command keys any time you go into the Finder, such as at bootup or out of *Waystation*. I don't have a timing on this, but it's a definite improvement. Maybe 2 to 1.

Crashing vs. ShutDown

The explanation I have heard is this: When the Mac opens a disk or hard disk, it marks a special location saying: "I've opened this disk". If you crash or turn off your machine other than thru ShutDown, that mark is never updated with "I've closed it". When you boot again, the Mac first looks at that mark: If it says 'Flameout!', then it has no way of knowing what might be damaged on the disk. In the PC world, this may result in the rest of your hard disk (all 18 Megabytes) being taken up by a mysterious half-open file that was never properly closed. I much prefer the Mac's way of doing this: It proceeds to check the entire desktop for 'litter', and fixes what it finds.

Thus if you habitually 'turn off' your Mac, instead of Shutting Down, you will have to wait while the checking and rebuild occur each time you boot.

Waystation

The Finder has its place, but it's a big program, has a lot of work to do, has to look everything up each time, and then build this wonderful graphic display to show you all the possibilities. It's similar to going into a disk utility like XTree on the PC instead of directly into the maligned A> (or C>) prompt. (Hey, it's fast!) Granted we don't want to revert to A> on the Mac, but there's something much quicker than the Finder: *WayStation*. It's a publically available 'minifinder' that allows you easily to install dozens of applications on a very quick menu. Extremely quick. On my machine, I leave an application and am back into *Waystation* in 1.2 (that's right 1 point 2) seconds. Launching is also fast (there's nothing to put away, no trash to empty, etc.). As in Apple's Minifinder, you cannot copy or delete files (unless you add a desk accessory that can), but you do have desk accessories (unlike the Apple Minifinder) and you can install applications from ANY folder level or even from other disks (also unlike Minifinder). You have direct access by a single button to the Finder or any 'Other' application you haven't installed, and it's much easier to install and remove one or many applications at a time than with the Minifinder. However, it only accepts applications, not documents, as entries in the menu. You can get around this by either going to the Finder, or by installing a renamed Minifinder (which will accept documents) as an application under *Waystation*.

Switcher

Switcher (and eventually, I hope, *Servant*) is the deluxe timesaver on the Mac. There have been similar things available on the PC for several years (MemorySwitch, DesQ) which unfortunately don't get the attention they deserve. Most people have at least heard of the *Switcher* on the Mac, but I still find people, for whom it would be perfect, trundling in and out of programs thru the Finder. I recently installed *Switcher*, *RamStart*, and *System /MacWrite*



/MacPaint on a MacPlus for a client. We booted into the ramdisk, which copied the applications and all system files, plus a *Switcher* set to load both *MacWrite* & *MacPaint*. When I had her double-click the *Switcher* set, and it opened into both *MacWrite* and *MacPaint* in about 3 seconds, she literally fell over with her mouth open. She had gone from the stone age (old floppies, old ROMs, old System, no ramdisk, no switcher) to a premier system in one day. Future shock indeed.

Minifinders under Waystation

The last time I checked the Minifinder, all files you installed in it had to be at the same folder level and on the same disk, a severe handicap for me. It also limits you to 12 items installed. But it is very fast, much like *Waystation*, except for lack of access to desk accessories. And it accepts documents as menu entries. So the way I use the Minifinder now is to get around the limitation of *Waystation* on launching documents. I put all the documents of related documents I want to use frequently into a folder together (my standard practice anyway), select them all, and then Install Minifinder. Then I rename the Minifinder with *FEdit* or something equivalent (You can't do it in the Finder; the name is locked.), move it into the System folder, and install it as an application under *Waystation*. A single click from *Waystation* takes me instantly to an organized menu of the documents; open any one of them and I'm into my document in it's application. Very fast. When I'm done, it goes back to *Waystation*. Even faster.

Using Ramdisk

Ramdisk, as you know, is memory emulation of a disk drive. It's about as fast as anything can get on the Mac (with the exception of the ROMs). You must have enough memory to use it, but even on a 512K machine, if you have one thing that you know you'll use all the time, it can pay to put it into a small ramdisk and run it from there. Things are 10-20 TIMES (remember that's 1000-2000%!) faster there. But it only pays to put things there if you are in fact going to be going in and out of them all the time. A trim System is one good candidate. Especially if you don't have enough drive space. With the System files in ram, everything runs faster. If you've got 1 or 2 Meg. The difference between ramdisk and cache is control. You decide what goes and stays there, not the cache software, which, depending on its intelligence, is supposed to be making guesses about what you'll need next.

Cache

To use or not to use: cache memory is a judgment call. Caching, as several have pointed out, keeps copies of disk sectors you have recently used, on the hope that you will need them again. It's using what you just did to try to guess what you're going to do next. After all, it hardly matters what files you WERE using. Big deal. The real value lies in knowing what you WILL be using. And therein lies the rub, friends.

The value of cache memory varies greatly with:

- The intelligence of the cache software
- How big it is in relation to the files you are using
- Whether you are going to be needing the same sectors again
- Whether you need lots of disk access or lots of memory
- Whether the cache size is fixed or can vary as needed

For instance, if the cache swaps out your frequently used font sectors because you load in some big file once...or if you run out of memory in an application because the cache didn't have enough sense to get out of the way...or if the cache brings in only a small chunk at a time when you're going to be reading the whole file, thus slowing down disk access.... As an example, if you

are simply reading sequentially thru a file (and thus will only be using each sector once), then the cache, which requires read-from-disk, use-in-program, where-will-I-cache-it?, maybe flush-from-cache, save-in-cache will actually slow things down compared to read-from-disk, use-in-program. (See loading of Waystation in ROM speeds chart above.) The same thing can happen if the cache is too small. This happens to me all the time. You're spending all your time replacing the cache instead of using it. That's why it depends on what you are doing. Tests like "First time to open *MacDraw*: 23 seconds; second time 7 seconds" or whatever are fine if that's what you're doing, but if you're using integrated tools, like *MacWrite* / *MacPaint* / *MacDraw* / *Excel* to write a report, you're better off using *Switcher* and turning the cache off.

I admit to knowing about cache in general, and not how the Apple MacPlus or Nevins (Are there others?) cache software works internally. Anybody have any more experience? Any numbers? The Nevins cache claims to be smarter. Anybody know?

Double-sided disks

	New 800K Drive		Old 400K Drive		
Initialize	800K	51	400K	35	
Erase	400K	30	400K	40	25% faster
	800K	55			
Copy <i>MacDraw</i>		25		29	13% faster
Load <i>MacDraw</i>		14		17	17% faster
Open Finder/2 Folders		6		9	33% faster

Twice as much space, about 25% faster. Comes with the ROM upgrade, all for under \$300. Need I say more?

2 Meg Memory Upgrade

There are several upgrades to more than standard memory now. I bought this one as a kit (\$675) and installed it myself over a year ago (not something I would recommend to everyone, but there was no problem), giving myself 2 Megabytes of contiguous memory usable by BOTH programs and ramdisks, PLUS a 30% speed increase in most processing. This was one of the best purchases ever. I would argue that for most people, 2 Meg of memory, to switch between applications is more valuable than a hard disk, and should be your first upgrade.

Word in the user community is that Levco is one of the smartest and most reliable companies. There are other new upgrades appearing now, such as the Three-in-One Touch Board, which give you even greater speed increase. Of course, there are things like the Levco Prodigy, which will run rings around anything else, but the price is steep. We're talking about average people here.

Launchers

Several Desk Accessory (or even one that is a Function Key) programs are now available that will let you launch another application directly from inside your current document. Some are in the public domain, such as the one on *MacQ*, others are commercial, such as *Dreams of the Phoenix's*, and Cortland Computers' *Launchers*. Cortland's, for example, lets you set the Quit ahead of time, so that whenever you quit, the new application will be launched. Since you avoid any type of Finder altogether, you save seconds every time.

Other DA

Other is a Desk Accessory that allows you to open other DA's without having to install them in your menu using the Font/DA Mover. *Other* allows you to speed up your System generally by not having as many DA's installed, thereby reducing System file size. It takes longer to select



the DA you want, of course, but for infrequently used DA's, or even for having access to a very wide range, *Other* can't be beat. That's why it's one of the DA's in my System.

Hard Disks

Hard Disks are a whole other topic all by themselves. Needless to say, increased on-line storage, faster disk access and so on are a great advantage. However, if you're not careful, you can make things slower with a hard disk than with a lean, mean floppy. There are some occasions on which I simply do everything on floppy disk. For instance, bringing up alternate Systems, working with ramdisks, doing development work when I'm not sure what I might do to my hard disk, experimenting with new programs...

But in general, my opinion is that AFTER large memory, a hard disk is the next most valuable addition, and of course other people put hard disk before memory. You pay a certain price in terms of reliability with some brands. And of course backup becomes a very real issue.

But for speed....Following are some indications of speed increase with the Apple HD20, which is among the older, slower non-SCSI drives. SCSI drives are even faster, in some cases fast enough to be amazing.

SCSI

Small Computer System Interface hard drives are even faster than the old style, although perhaps by not as much as you might think. Most of the reviews indicate 30 - 50% increase. Some of the newer methods of increasing the speed of the SCSI interface itself, such as the one for the DataFrame, which plugs into the ROM socket inside the Mac, have been promised to double the transfer rate.

Working Intelligently

Making smart decisions about what to use when increases the effective speed of your Mac. Use only one system on the disks, for instance, unless there is a reason you want to be switching Systems each time you load a new application. Make some decisions among whatever you have available: ramdisk, *Switcher*, Finder, Minifinder, tailored floppy disks, and hard disks. Make backups. Save printing jobs, print all one type of output as a group from the Finder, use a spooler, or print later when you're done with the Mac. Keep your powder dry.

Hockabout pict

Why Everyone Should Get The New ROMs

Gordon Haig

If you have a basic Mac — a 512k floppy system - and are using basic software - *MacWrite*, *MacPaint* — is there any reason why you would want to get the ROM upgrade? Especially at a cost of 250 or 300 dollars. For that much, you could buy a modem (from BMUG), and have enough money left over for a couple of new programs, or a reconditioned 400k external drive. If you have a 512 you apparently have lots of hardware power for programs like *MacWrite* or *MacPaint*, as they would run on a 128k. What's the real value of, say, the increased speed in reading from the disk drives? After all, 30 or 50% improvement sounds great, but it won't increase your typing speed. One can't help but wonder if it is all just a bit of hardware hype.

I operate a 512k Mac with 2 floppies. I've had the new ROMs and the 800k internal disk drive for about 3 months now, and I'd like to explain why the upgrade is worth considering.

The particular value, as I see it, is that the disk drive and ROM upgrade fulfill and complete the existing potential of the Mac. Rather than having some new, wonderful, but possibly unnecessary capacity, your Mac just does everything that it did before, faster, more easily and less obtrusively. And therefore your work goes faster and more easily, and the Mac supports you without constraining you. If you are going to spend any money at all you should consider this.

The principal differences in operation are that the disk drives are faster, operations that take place within a program are faster, the disk cache can be used to keep things you may need on tap in RAM, and the 800k drive allows you to have your whole working set up on disk, in the drive, and immediately accessible.

Because the disk drives work faster, applications launch more rapidly, documents load with minimum delay. You see the wrist watch less often, and for shorter periods. Because it's based on ROM changes, the speed up includes your existing 400k external drive.

The greatest difference in working with the new ROMs is the way they speed up the moment to moment operations of the Mac within a program. Text will now scroll as fast or faster that you can read. If you change the existing font in a document, or change the ruler, the document reformats much more rapidly. For large cuts or pastes, for a jump to another section of the document, for whatever process, the screen now redraws much faster. If you resize the window, or click back and forth between a desk accessory and your main window, hardly more time is taken than needed for the actual redrawing of the screen. Operations that were slow enough to drag the action on the old Mac are fast with the new ROMs. That hesitation, with nothing happening on the screen while the Mac figures out what to do, has disappeared.

If you set the cache on the control panel, parts of the system that you use often, like dialogue boxes, fonts, and DA's, are held in RAM and come up instantly. This complements the speed up of ROM based operations, by further speeding up disk operations, and eliminating many that take place within programs. For instance, in disk-based *MacWrite* all or part of the file is held in RAM and when editing you don't have to wait for the spin of the disk to bring up the next section that you want to look at.

All these changes contribute to the actual speed of the machine. The 800k drive contributes to the ease of your part of the job. With 800k on the disk it easy to keep a whole project's files on



one disk, or to keep two applications you use together along with their files and the system all on one disk, or to get the maximum enhancement of your basic applications by installing a full range of desk accessories and fonts.

The larger drive minimizes those most exasperating of MacOperations, "Please insert the disk: " Or the common, though undocumented, "Where the !@!!! disk with my file on...." With the 800k disks it is much easier to set up a function-oriented disk, with everything you might need for a particular kind of project on the disk. Say *MEdit*, *MacWrite*, your system files, and a full complement of desk accessories, including *Cheap Paint*, *MockChart*, *MockTerminal*, *MockWrite*, *Art Grabber*, *SkipFinder* or *Transfer*, *DiskInfo*, *Key Caps*, Control Panel, Chooser, and *Idle*. Integration on the cheap. Then let *SkipFinder* do the searching for that file. All without sacrificing fonts or having to hack the system file, and with your external drive, if you have one, still free for additional data files.

So What?

What effect do these changes have on the overall feel and process of your work? Let me take basic word processing as my example.

Consider word processing as a video game. It allows you to move things around on the screen and make things happen to them, and experience the results of your actions instantaneously. Just like *Space Invaders*. That is one of the really wonderful things about word processing: that the representation of the text fully transforms itself to reflect your changing thought.

But now consider *MacWrite 4.5* (with the old ROMs) as a video game. What kind of game is this? Just when you start to get into the game, you push the firing button the wrong way, and all the action freezes for 30 seconds.

Now *MacWrite* is not in truth a video game, or at least not only a video game, but it does have a pacing problem. Productive work, like playing a good game, has a rhythm. If you are working steadily and productively you develop a rhythm, and the rhythm will in turn help you sustain your continued concentration and momentum. But *MacWrite* interrupts your rhythm. MacWait.

The special value of the new ROMs to someone doing basic word processing, is that they solve the pacing problem. The speed of operations has now passed that critical point below which delays no longer obtrude themselves on your attention, nor interfere with your next impulse to write. You can maintain your focus on your work and continue to work, supported and assisted in the natural workings of your thought, and allowed to concentrate fully on your subject. You are not forced to become aware of the Mac's operations nor to allow for "work arounds".

The new ROMs then allow the Mac to more completely support you, and to achieve more fully those particular excellences that are its distinction and our pleasure. Basic functions, like word processing, work better, and the Mac is more Mac-like everywhere.

BMUG
FALL 1986

I'm Upgrading!

Ken Wait

Goodbye to you my 128K Mac,
Though after you go I'll have quite a lack,
Of disk swaps, and this pain in the back,
(of my hand).
You turned us all on to the Mac idea, it's true,
But now it seems to me we've forgotten you.

You had just one Finder, one System too,
But then, every program could use every you.
The spreadsheets, word processors, and databases were squashed,
So that all would fit, into our Macintosh.
It was true, every single one matched,
Until the day, a second drive could be attached!

Soon the 512K was the Mac of the hour,
Still 100% compatible,
Nothing gone sour.

Programmers and Developers began to take note,
While 128K owners were denied a vote.
Soon there were several different 1 meg boards,
Then 2 megs, 4 megs, K by the hoards!
But who was to set the standard, who with it would grapple,
Would it have been, could it have been, you know who?

Soon Apple the 128K stopped producing,
I guess they assumed we would just stop using...

But this is not where my story does end,
For still with my friends I must contend.
For each has his upgrade, a different model they've got,
Each in its own way the best of the lot.
Some software runs in John's, some runs on Jack's,
But less and less of them run in my Mac.

So goodbye to you my 128K,
But before I upgrade I'd just like to say:

You showed us the beacon, you played well your part,
For in that lies the beauty that thou art.

Ken Wait is a Santa Cruz student who works for a Bay Area computer retailer during the summer. One of his main hobbies is staying up until 2:30 in the morning, writing poetry as if possessed, which he probably is.



The heart of Apple II is still beating

Copyright ©1986 Linda Custer

On Monday morning, September 15, 1986, the Apple IIGS was officially announced to an eager press. Just as the Macintosh represented a radical departure from the current standard in desktop computing and defined a new standard, the IIGS brings the best of that technology to maturity and popularizes the standard. Yes, the heart of Apple II is still beating, but now it has had a little coaching from its precocious younger brother, Macintosh.

The soul of the new machine

The first thing you'll notice about the IIGS is that it doesn't look like a Mac or any Apple II that came before it. The IIGS board and power supply are in a low box, not much larger than an HD20. The disk drives, keyboard, mouse, and monitor are all separate, and can sit to the side or on top of the main unit. Unlike most other computer systems that come in parts, however, the IIGS has a desktop footprint that is only slightly wider and deeper than the Mac's. The IIGS disk drives and monitor can stack easily. Although the IIGS footprint is similar to the Mac's, the IIGS just isn't as transportable since it comes in so many pieces and has a large monitor.

The next thing you'll probably notice is that the IIGS clashes with your beige Mac. The IIGS is platinum, a new, rather pleasant grey color which Apple has decided is the color they will use for all their future computers. Its plastic case is similar in quality and texture to the Mac, but its air vents cut into the sides and top look more like those on a //c.

Next, of course, you'll want to look inside the case. Another surprise: you won't need to get out the Torx screddriver. The IIGS doesn't contain one screw! Instead, all its parts simply snap together. Once inside, you'll find that the IIGS sports a 65SC816 chip. Although the 65SC816 is not found in any other Apple product, it emulates the 6502, the processor in all other Apple IIs, perfectly. Old Apple II programers should feel at home with this chip. Its top speed has been bumped up to 2.8 MHz from the old 1 MHz speed.

The 65SC816 is in good company, too. The IIGS contains 7 custom chips to help perform special functions and handle some of the load. Probably the most important of the customs is a chip called the Mega II. It has all the logic of an Apple //e or //c in just one CMOS IC. Now that's miniaturization.

GS: Graphics and sound

If you like the Commodore Amiga and the Atari 520 and 1020ST, you'll be happy to hear that the IIGS contains special coprocessors to help with graphics and sound. (After all, remember the 'G' and 'S' in the name.) The video chip, called the VGR (video graphics controller), is not quite as advanced as the graphics chip in the Amiga, and does not support graphic "sprites", images that can quickly be drawn anywhere on the screen. However, it can automatically fill any rectangle with a color, and works well with analog RGB monitors. The sound chip, by Ensoniq, is the most advanced in any peronal computer. It can asynchronously play up to 15 voices, and can produce voice-quality output. Ensoniq is already famous for making chips for some of the best musical synthesizers, and now they have brought that technology to a personal computer.

Besides the 'GS' chips, the IIGS also includes custom chips to control the keyboard input, the disk drives, the slot addresses and functions, and communications between the CPU and the Mega II chip.

Port-able

Apple II users will recognize the seven ports inside the IIGS as standard, old-style Apple II ports. Almost all cards used by the older Apple IIs will work fine in the IIGS, although some "phantom" cards will create problems because they act like they occupy more than one port. Of course, brand new cards will also be developed for the IIGS by third parties. Apple hopes that among these is a card which will allow the IIGS to emulate an IBM PC.

An eighth, smaller port in the IIGS is reserved for future memory expansion. The base IIGS comes with 256K of RAM and a 128K ROM chip. However, the 65C802 can address up to 8 megs of RAM and 1 meg of ROM, so memory gluttons should be happy, at least for a while. The IIGS ROM contains QuickDrawII and a memory manager similar to that found in the Macintosh. Other Mac-like toolbox routines reside either in this ROM or in code that is loaded from disk with the system.

New to the IIGS are two RS232, subminiature DIN-8 ports identical to those found on the Macintosh Plus. In most cases, printers, modems, data acquisition devices, and AppleTalk networks will connect directly to these ports. Printing, networking, and telecommunications software should know how to open and control those ports. The IIGS is even capable of starting up over an AppleTalk network. Rather than having to use a system floppy or hard disk to begin working, you can start the IIGS over a network that includes a file server. The supporting server is not yet available, but Apple is expected to announce one in the near future.

One thing you won't find in the IIGS is a fan. Remember? Apple doesn't want you to hear the computer when you're doing your best work late at night. Most of the IIGS chips use heat-efficient CMOS technology, so you may not need a fan. However, Apple does offer a IIGS system fan as an option, though they recommend that you install it if you fill three or more IIGS ports with cards.

The key is the bus

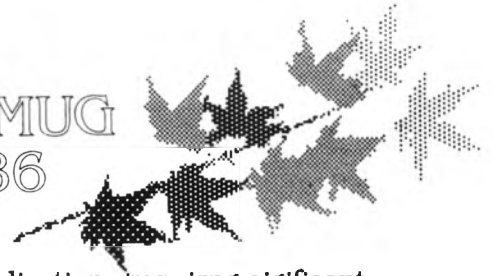
The IIGS keyboard is as stylish as the computer itself. Its keys are layed out similarly to the MacPlus keyboard, and it includes a numeric keypad. However, it is thinner and lighter than the Mac keyboard, and has a more futuristic look. Its feel is similar to the current //c keyboard. The IIGS mouse is a bit smaller than the old Mac mouse. It is also shaped a bit less like a box than the Mac mouse, and has a more comfortable feel.

The IIGS is the first computer to sport Apple's new standard input bus, called the Desktop Bus. One port on the back of the IIGS attaches by cable to the keyboard. Then the mouse, and any other input device—such as a MIDI keyboard, a graphic tablet, or a game controller—can attach directly to the keyboard or other adapter on the bus.

Drive it away

Apple chose to make the "Disk 3.5" the standard IIGS drive. Although UniDisks will also work, Apple's new platinum disk drive will be the standard for all future Apples. The Disk 3.5 uses something Apple calls SmartPort technology, so it needs no controller card. It also, fortunately, includes a red drive light and a disk eject button. (By the way, the Disk 3.5 also works on the Mac.)

Apple II owners who already have 5.25" drives can use those drives in the IIGS, as long as they put their drive controller card in one of the slots. The only disadvantage of using the older



drives is that their capacity may not be large enough for many applications requiring significant amounts of graphics and sound memory and they operate more slowly.

Apple is also marketing a SCSI card for the IIGS. This card implements the same SCSI interface as is found on the MacPlus. To go along with the IIGS announcement, Apple announced a 20 meg SCSI drive, called the Hard Disk 20SC. It comes in platinum or beige and works on either the IIGS or the Mac. (Partitioning, not yet available, will be required for the IIGS and Mac to share the same drive.)

Monitor the situation

The IIGS really shows off when it is connected to Apple's new analog RGB monitor. Although the IIGS can be used with a composite color or a monochrome monitor, its superb color graphics only look their best on the analog color screen. Also, the analog color screen requires no card to display its graphics, since the controller is on-board.

The IIGS supports many graphics modes. The old Apple II 40 column and 80 column text, lo-res, hi-res, and double hi-res modes work as always. The text-only modes have three user-assignable "areas", each of which must be only one color: the background, the text itself, and the screen border. A new video mode, dubbed super-res, works over the analog RGB monitor at either 640 pixels horizontal by 200 vertical or 320 horizontal by 200 vertical. Although many programs will make use of the crispness of the super-res mode, the extra resolution comes with a sacrifice: on each scan line, only four colors can be used when displaying 640x200. Sixteen colors per scan line are possible with 320x200 resolution, and thus most "paint" programs will choose this mode. Since only 320 dots per line are available, any Mac-like menus in this mode are limited to about 40 characters across, and do not look as sharp as the menus Mac users are used to.

It's about time

The IIGS has a battery-backed up parameter RAM and a clock chip, like the Mac. The parameter RAM records port, display, and international settings so these do not have to be reset each time the machine is used. And for those who still have their Apple II game controller joysticks, yes, the IIGS still has the good ole game port.

But does it do icons?

The Apple IIGS operating system is a variation of ProDOS that works well with the 65SC816, called ProDOS 16. Without even inserting a disk, you can operate in AppleSoft, like the old Apple IIs, with a command line. The IIGS also supports a control panel on power-up, although until Apple's IIGS Finder is released, it is text-only. The control panel allows you to make port settings and control the text, background, and border colors for text mode. It also allows you to control the clock chip settings so you can force the IIGS to operate at 1 MHz and have at least a chance at winning your favorite old games.

The IIGS runs almost all Apple II programs with no modifications. Heavily copy-protected products that rely on critical timing may not work, and some communications programs which access cards directly may have problems working with the new SCC chips, but IIGS-compatible updates to those programs should be available soon.

Along with the announcement of the Apple IIGS, Apple announced AppleWorks 2.0. The retail price of AppleWorks remains unchanged, and current AppleWorks owners can upgrade for a small fee. The new AppleWorks features mail merge and more efficient operation, as well as a host of improvements throughout the program.

This little Apple went to market

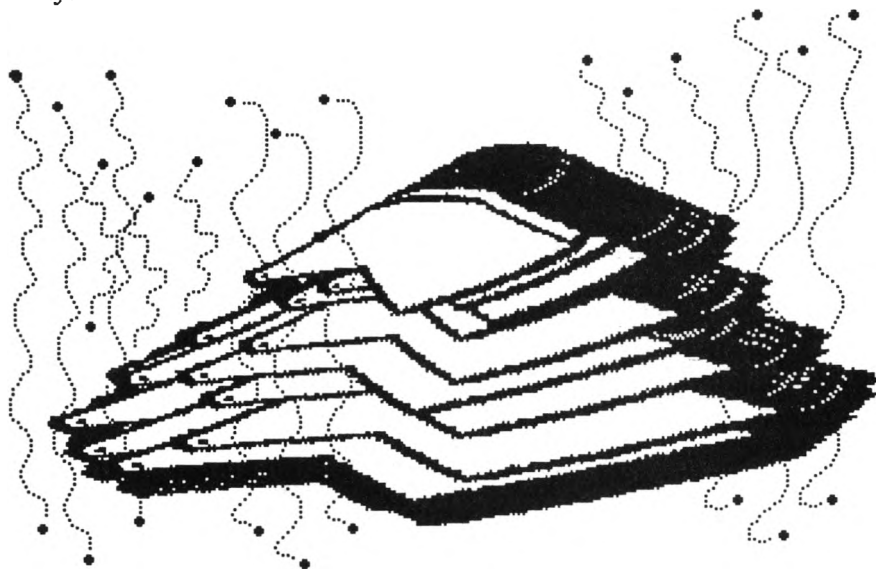
Apple will not represent the IIGS as the complete solution for everyone. The //e is still very much alive, and a brand new memory enhancement for the //c—allowing it to go to 1 meg of RAM—was just announced. Apple will present the IIGS as a high-end computer for the educational (K-12) and home/consumer market, while the Mac will be billed as the college and business computer. Wherever the //e or //c is now, if better speed, graphics, or sound is needed, the IIGS will do nicely. But for power spreadsheets or databases, or for serious desktop publishing, the Mac is still the one.

Interestingly enough, the Mac has been the favorite computer of many professional musicians, but the IIGS may change that. The one area where the IIGS has it even over the Mac (even though the Mac is great) is in synthesizer-quality sound. The IIGS has digital sampling built in, and can play fifteen voices asynchronously to the Mac's four. Apple even went as far as announcing special Bose speakers with the Apple label to plug into the IIGS!

Ads for power users

We will see some very interesting advertisements from Apple beginning with this year's World Series. BBDO, Apple's new advertising agency, has a strong ad campaign planned to sell the IIGS to parents. Since the Apple II has always done well in grade schools and in the home market, Apple hopes the IIGS will strengthen their stranglehold on that market. The new ads feature children learning everything from baseball to zoology, with the song, "Teach", playing in the background, and with Apple II computers showing up every so often. The moral of the commercials is: Apple. The power to be your best.

Apple's newest addition to their product line demonstrates their commitment to merge the Apple II and Macintosh product lines with a common base of ROM routines and a common user interface. It also demonstrates their desire to have one common set of peripherals and connectors. We should expect future Macs to look more like the IIGS and to include some of the advanced color and sound capabilities as well. If you want a glimpse of the future, look at the IIGS today!



Race/Hockabout

Radio Shack Model 100 Portable Computer

Gordon Haig

Introduction

If you travel or work at more than one location, a Radio Shack Model 100 or 102 (also known as Tandy 100, 102) laptop computer is the single most valuable and versatile hardware addition available for your Macintosh. With a few minor exceptions as noted below, the 100 and 102 are the same. In this article we'll use the 100 as a generic designation for the 100/102 series.

For about 500 dollars you can do basic word processing or data entry anywhere and stream the work into your usual computer environment without interruption or inconvenience to you. It's like having a portable keyboard, a terminal for your Macintosh that lets you use the Macintosh wherever you are. Take notes on the Model 100 in a conference, in the library, on a trip, or some place in the sunshine. When you get back to your Macintosh, plug a cable into the Model 100, transfer your file to the Macintosh, and continue working.

A laptop gives you connectivity: you type in your thoughts as you develop them, extend them and transform them; pour them into the Macintosh. You never have to transcribe, retype, or squint at scribbled notes again. Whatever you do merges as seamlessly with the main flow of your work as if it were done on the the Macintosh keyboard itself.

Macintosh users who use a Model 100 include:

- a teacher who uses the Model 100 at school to record marks and outline lesson plans for later uploading to the Macintosh
- a graduate student who uses it as a combined bibliographic file and 5 by 8 notecard replacement for library notetaking
- a consultant who uses it as a central appointment and address book and as a handy file transfer medium from his Macintosh to other locations and types of machines
- a reporter who uses it as a combination typewriter and portable terminal to phone in his reports.

The Model 100 is also used as an order book and expense notebook for salesmen, as the brain of test instruments or input-output monitoring devices and as an extra keyboard in offices where they don't wish to purchase another full sized computer. Almost the whole spectrum of standard character oriented applications is now available, as are innovative applications that take advantage of the 100's portability and durability.

The Model 100 is equipped as a superior notebook that combines writing space with address book, appointment and expense records, and has report-from-anywhere communications capability. It has a good keyboard and a straightforward text editor with keyboard equivalents for all commands. The screen text is quite legible, much more so than on other portables. You can write, reread, and then extend your thoughts with fluid ease. However, although it is a superb instrument for writing a short report or drafting a longer one, the Model 100 has the limitations of a notebook as well as the advantages. The size of the display, 8 lines by 40 characters, is equal to about five lines from MacWrite. When it comes time to reorganize, to

rewrite in a finished style, or to print a professional quality document, you'll want to be back at your Macintosh. For this, the Model 100 has convenient communications capabilities, and the facilities of the Macintosh desk top environment allows you to incorporate the use of the Model 100 as if it were made for the Macintosh.

The Radio Shack Model 100 has an 8 lines by 40 character screen. The screen is very readable for a portable as the letters look like a wider version of 18 point Monaco, though with the lines less than single spaced. Screen is 2 x 7 1/2 inches: viewing angle is adjustable using a dial on the side of the machine.

The Model 100 itself is a flat box about 8 1/2 inches by 12, 2 inches thick. The keyboard, function keys and screen are all part of the top surface of the box. The 100 weighs 5 pounds and is easy to carry, but not weightless. The 102 weighs in at 4 pounds and its keys are slightly higher than those on the 100. Both run on 4 AA batteries or an AC adapter. There no moving parts except the keys and no vents or openings into its interior; both are very durable. The screen, made of soft plastic, is a partial exception: store or carry the laptops in the case supplied. Do not, however convenient it seems, put anything else in the case; especially not papers with staples.

Both Models come with 24k of RAM and 32k of programs in ROM. The programs include Text, a simple word processor; Telecom, a communications program; Address, which stores names, addresses, and phone numbers; Schedule, which has simple database capabilities good for storing appointments, mileage, and expenses; and an MS BASIC interpreter. The Model 100 comes with no built-in mass storage except for its memory. Files in memory are not lost when you turn the power off; they are maintained by the batteries and by a built-in Ni-Cad back-up. For the programs in ROM the original 24 K is all data space and is adequate for day to day use in conjunction with the Macintosh.

Using the Model 100

Using the Model 100 is easy for an experienced Macintosh user. Basic word processing on the Model 100 works almost the same as on the Macintosh: open an existing file or the text program itself, and type. You can cut, copy, paste, and select, text. Select by pressing the F7 function key at the top of the keyboard, then move the cursor from one end to the other of a block of text; the selection is reverse highlighted to indicate. Of course you must access these functions using arrow cursor keys rather than with a mouse, but the functions of the system are almost identical, and the labeled function keys do not require memorization. There is one notable difference—no scroll bars. When you move the cursor off screen, the display "chases the cursor" to show it at its new location.

The Model 100 's benefits are apparent when you transfer data into Macintosh programs. The ROM programs, plus the Sort program listed in the manual, all generate or use ASCII text files, and programs from the MS BASIC interpreter can be saved as ASCII files. Any file from the Model 100 can be opened easily in MacWrite, Word, or a desk accessory text editor, or in other programs, such as databases that allow you to import text files.



How to Move Your Files

To transfer files you need a null modem cable and a communications program for the Macintosh. An easy to use communications program, Telecom, comes with the Model 100.

The null modem cable must plug into your Macintosh at one end and have a standard male RS-232 plug on the other. The Imagewriter I cable is ideal. If you must buy a cable be sure the it will fit into the narrow slot for the RS-232 port on the Model 100. Clamp-on connector hoods, such as those used for Radio Shack solder type connectors and for some commercial cables, are too large. Once you own a Model 100 and transfer files regularly, you may want an A/B switch box and a cable especially for the Model 100, but this is not essential to start.

Any communication program on the Macintosh will do; the simpler the better. A desk accessory terminal program is most convenient for everyday use: you can get one as part of the Mock Package, Sidekick, or Quick and Dirty Utilities. They are simple to use, and often come packaged with a DA editor that can open Model 100 files. I use MockTerminal 4.2 and keep a copy of it as part of the system file on my MacWrite disk. That way I can boot MacWrite, load my file from the 100, and go directly to work.

Move Your Files To The Macintosh

To move your files, start with both Macintosh and Model 100 turned off. Plug your cable into one of the serial ports on the Macintosh and into the Model 100 RS-232c port. If you are actually using your Imagewriter I cable you need only unhook the printer end of the cable and plug it into the top of the Model 100.

Turn on the Model 100: Telecom appears on the main menu. Write down or note the exact name of the file you will send to the Macintosh, as it appears on the menu. Include the ".do" extension. Move the cursor to highlight Telecom and press ENTER. A new screen will come up with the telecom prompt and last parameter setting at the top, and on-screen labels for the function keys across the bottom. To set the telecommunications parameters we will use F3 "Stat." Press F3, and the prompt, also "Stat," will appear on the screen followed by the blinking cursor. Type in 68N1E<enter> (that is, 68N1E followed by pressing the Enter key). If you then press F3 again, followed directly by <enter>, the new setting will print to the screen. It should now be 68N1E,10pps which stands for:

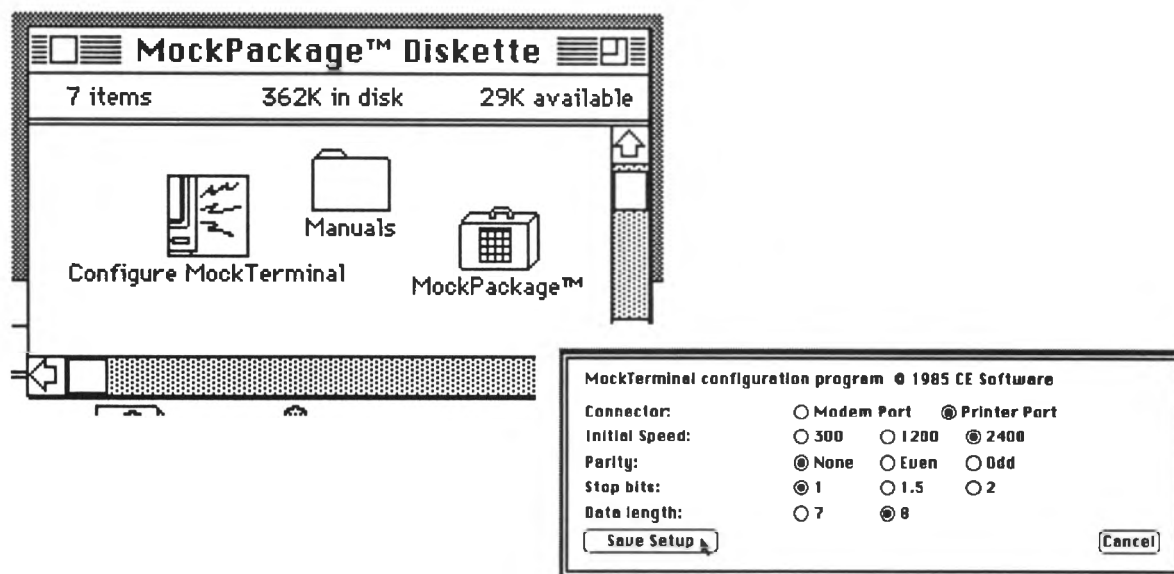
6	2400 baud
8	8 bit word
n (case doesn't matter)	no parity
1	one stop bit
e	Xon/Xoff enabled.

The setting after the comma refers to the dialing rate when using the modem, and can be ignored. If the overall setting is not correct, press F3 once again, retype the setting, and <enter>; and recheck.

Set the communications parameters on the Macintosh to match these. If you are using MockTerminal, turn on your Macintosh and boot up with a system disk that includes it. MockTerminal has an associated utility application, Configure MockTerminal; while you are on the desktop, double-click on Configure MockTerminal to bring up the configuration dialog box. Set the radio buttons to 2400 baud, 8 bits, no parity and one stop bit, and select the appropriate serial port, the printer port if you plugged the Model 100 into your printer cable. Xon/xoff is automatically enabled and you need not set it. Click on "Save setting" and quit; then call MockTerminal from the Apple menu. A MockTerminal window will appear and with

BMUG FALL 1986

its own menu heading to the right of the menu bar. Choose "Start Recording" from the MockTerminal menu, then enter your file name in the dialog box and click on "save." The Macintosh is ready to receive your file, and anything sent from the Model 100 will be saved to disk.



On the 100 press F4, "Term:" this enters terminal mode and changes the on-screen labels of the function keys. F4 should read Full for full duplex; if it reads Half, push it once to toggle to Full. Press F3, "Up." The Model 100 will prompt "File to Upload?" Type in the exact name of the file as it appeared on the main menu, followed by <enter>. The next prompt is "Width?" for line width. If you are using MockTerminal just press <enter> and the upload will start. (Terminal programs other than MockTerminal may require you to type in 80 <enter> to insert carriage returns).

The one thing to be absolutely sure of in Telecom is that you do not mistakenly press F2, Down[load]. This can get confusing, as the idea of downloading to the Macintosh seems to make sense. Pressing download = No Warning + Instant Disaster = Total Loss. The Model 100 will download empty space into the file formerly occupied by your text. This is a particularly unpleasant experience because it is nearly impossible to lose your work on the Model 100—files are saved automatically, and are not lost when the power is turned off.

To the Model 100, the Macintosh is so big it looks like a mainframe. Other people write "Send" on a bit of self-adhesive label and attach it above F3. You may also wish to make a back-up copy on the Model 100 of really important files before sending them. Use "Copy" in the Text program. Select the whole of your text, copy, then open a new file and paste into it.

When the last of your file scrolls onto the Macintosh screen, press F8 "Bye" on the Model 100, answer "Y" to the "disconnect?" prompt, press F8 once again to return to the main menu, and then switch off. On the Macintosh, select "Stop recording" from the MockTerminal menu, and then click on the close box. On the desktop you will see your text file.



Telecom and MockTerminal make the file transfer process easy even if you are unfamiliar with the communications process. Once you have set Stat and made one successful file transfer the Model 100 will save the correct setting. For further transfers go directly to Term mode and upload. Install a copy of MockTerminal on your Write or Word disk, leave it appropriately configured, and you can start up directly to MacWrite, upload your file, and begin work, without ever looking at the communications parameters. The one remaining inconvenience is remembering and entering the often cryptic file names, and this can be eased by entering a few lines of BASIC code that enable function key 6 to give you a listing of your files on screen while in terminal mode (see sidebar). The combination of Telecom and F6 is particularly convenient if you have numerous small files to upload.

Download

If you try to download to the Model 100 simply by reversing this procedure, you will find that it doesn't work. The Model 100 prints to the screen very slowly and cannot keep up with even a 1200 baud transfer; its effective top rate is 800 baud. It sends a very rapid series of Xoff signals to the Macintosh. MockTerminal does not respond to them fast enough and sends so many characters that the m100's buffer is overrun. Result: lost words and phrases from your file.

The best way to download is not to use Telecom at all, but rather the built-in Save and Load functions in Text. Set up the Macintosh, cable and Model 100 as before. Turn on both machines; launch Text on the Model 100; and type in and enter a new file name, as if you were going to start writing in a new file. Once in the empty file, press F2 "Load." When the prompt "Load from:" appears type Com:68n1e<enter>(with no spaces anywhere at all) The Model 100 is now open to receive your file through the serial port. On the Macintosh, call Mockterm from the Apple menu. Select "Play back ...", and a dialog box with available text files will appear. Select the file you wish to download, click on the "Open" button, and MockTerminal will send the file out the serial port to the Model 100.

Since the file does not appear on the Model 100 screen, you must depend on the Macintosh to tell you when the file transfer has been completed. The easiest way to tell is to listen to your disk drives: at 2400 hundred baud they will whir continuously until the file is transferred. If that is not convenient for your environment or hardware set-up, pull down the Terminal menu briefly. When the checkmark disappears from beside "Play back . . ." the transfer is completed. Do not hold the menu down continuously as that will halt the transmission. When the check disappears, type a Command-Z on the Macintosh keyboard and the Model 100 will end the load and print your file to the screen. Due to a bug in MockTerminal four or five characters will be lost at the end of the file, but this does not indicate any problems with the rest of the transmission. You are ready to turn off both machines, disconnect the cable, and take off with Model 100 and file in hand.

The Save function can be used in a similar fashion to upload to the Macintosh, as a substitute for Telecom. For day to day use by an experienced user, Save and Load are faster and more convenient than Telecom. To upload, enter your file directly from the main menu, press F3, type COM:68n1e at the "Save to:" prompt, and then (if the Macintosh is ready) press Enter. You need not remember or enter the exact 6 letter filenames, and there are fewer steps and commands required. In addition, for MacTerminal and some communications programs on brand X computers, using the save function is the only method that works—set such programs to receive a file transfer rather than to print to screen.

The principal disadvantage of this method is that it requires you to enter the communications parameters on the Model 100 each time you wish to transfer a file. If you are new to non-Mac

computing and to communications, fooling with the parameters can be a venture into the unpredictable. You might prefer to stick to Telecom and leave it set. If you need to get a file into the Model 100, FreeTerm can handle a download at 2400 baud.

Two miscellaneous items of information that will make your file transfers easier:

First, MockTerminal does true multi-tasking. If you are transferring a large file, you need not wait while it uploads nor continuously check the menu to see if a download is complete. Shrink the terminal window, click in the window of your main application to make it active, and MockTerminal will load the file while you work.

Second, if you get an A/B switch box in addition to your null modem cable, you will be able to connect the Model 100 to the Macintosh without turning off the Macintosh. (Under normal circumstances failure to turn off the Macintosh will cause intermittent and mysterious problems for the printer or modem port and might possibly damage the Model 100.) Leave the Switch box permanently attached to the modem port, and the cable to connect to the Model 100 attached to one of the ports of the box. With this set up you must turn off the Model 100 before connecting it but need not turn off the Macintosh. If your connecting cable is on the A side, switch the box to B, connect the Model 100, and then reswitch the box to the A side. Turn on the Model 100 and proceed. PBI Software makes a switch which is convenient for this purpose because its pushbutton allows one-handed switching.

The Model 100 As An Everyday Tool, Or What To Do With All Those Little Files

If you are short of time a laptop enables you to write where you are and when you can. If you have a report due, you work on it each chance you get, stealing a half hour one day, an hour and a half the next. Soon you'll have written a page or two to cover each of the key elements of the topic. However when you fashion a final piece you all too likely find that you have not one integral draft but 5 or 6 or ten separate but overlapping files. The discontinuity of your time is reflected in variability of focus and tone, as your memory plays you for the fool and your perspective alters from one day to the next; your "draft" is but a set of notes. How will you join these pieces into one?

Or perhaps you have written ten or twelve letters: how will you get them properly formatted and printed? To use these files you need to have more than one file on screen at a time, for convenience in consultation and in cutting and pasting. MacWrite, since it allows you only one window, is tough to use. Word is the program that offers this as a matter of course and if you have Word it will probably be your program of choice. But Word lists for nearly \$200, and you just paid \$400 for the Model 100. What can you use instead or until?

If you are using MockTerminal you will also have the other elements of the MockPackage, MockWrite and the MockPrint text file spooler. MockWrite will allow you to open a second text window in MacWrite, or to open a Model 100 file from within any program. In MacWrite this works particularly well for formatting letters written on the Model 100. Create a preformatted generic letter in MacWrite, followed by a page break. Copy the letter plus page break. Paste repeatedly until you have a copy of the generic letter for each letter you will send. Then you can successively open each file from the Model 100 in MockWrite, copy it, click to activate the MacWrite window, click again to set the insertion point, and paste to form the body of the letter. If you have recorded names and addresses in the adres.do file on the Model 100, or in a text file on the Macintosh, you can similarly open that file in Mockwrite and paste in the addresses.

In assembling that multiframe report mentioned above, your best choice at present is an editor

with multiple windows. If you use an editor, you will be able to open your files on the Macintosh, and later load them back into the Model 100 without converting or re-converting them. In addition, editors are fast, take little disk space, and add minimal additional expense to your purchase of a Model 100. MEdit, which was included on disk 40 with the Spring Newsletter, is an HFS compatible editor that allows multiple files open on the screen, will wordwrap each file to the size of its window, and takes up about 40k on disc. It will open files up to 32k in each window, so it can open any Model 100 file, and with multiple windows longer documents can conveniently be handled in subsections. In addition, if you need to massage an ASCII file before loading it into a database, MEdit will search and replace both tabs and carriage returns. If you have never had the opportunity to use a word processor with multiple windows try MEdit; the convenience it makes possible will be a delightful experience.

If you have ThinkTank 512, you have an alternate and powerful tool. Load your Model 100 files enmasse into text windows in an outline via ThinkTank's scrapbook commands. Name them, and split them up as necessary. By dragging the headlines you can very rapidly manipulate chunks of text, and thrash them into a logically coherent order. Think Tank is so powerful for structuring and restructuring your document that you are soon writing again and wishing that Think Tank was more powerful as a word processor. The ideal application for writing with the Model 100 would combine Think Tank's outlining function and ease of reordering, with the multiple windows and ease of writing of the text editors. I am currently experimenting with More.

Travelling with the Model 100

Travelling with the Model 100 makes you feel like a cowboy or an old soldier: you're travelling light, you don't need to worry about your equipment, and you can whip it out at a moment's notice. It'll fit in your backpack or flight bag, and when you turn on the power the main menu appears faster than light from a lightbulb. Expansion products should retain the light weight and uncomplicated portability of the Model 100; its true laptop usability, instant readiness, and easy operation; and its relative imperviousness to rough handling and or hostile environments.

However using the the Model 100 away from the Macintosh will rapidly bring you up against the need for some sort of mass storage. Notes you have finished, names and addresses you are storing, perhaps a utility program in BASIC, all are stored in RAM. A 24k Model 100 will hold the equivalent of almost 6 pages single spaced in MacWrite, with default ruler settings. About 10 and one-half pages double spaced. That's a lot of typing to do on the commute trip or a single plane flight, but not very much if you are away for several days and can't dump to your Macintosh. What are your options?

Get an additional 8k memory chip. After your Radio Shack warranty has expired get one from a third party vendor for \$25 or \$30 and snap it in yourself (e.g. \$29.95 from PG Design, at (313) 727-2744). Model 102 chips are cheaper, listed at \$14.95 from Radio Shack.

Use the modem to off-load. A direct connect cable from Radio Shack costs \$19.95. Send yourself electronic mail; store 128k on CompuServe (free with your account); or use a communications program on your Macintosh that allows for remote control, such Red Ryder or Microphone. Arrange for someone to boot the Macintosh, or leave it on and put a timer on the power supply to your auto-answer modem.



BMUG FALL 1986

In an emergency Radio Shack sells a cable that allows you to use most any cassette deck for storage. (15 or \$20). (A-R-R-R-R-R-R-R-G-H! Eat death, Tape Worms). Tape storage is neither light, nor durable, nor carefree, nor fast.

Now for the more expensive alternatives. Easy memory expansion on the Model 100 stops at 32k: after that it's bucks and bank switching. Around \$250 for the first added bank of 32k, up to \$425 to add three banks, for a possible total memory of 128k. However, it fits right inside the Model 100, has no moving parts, and increases the power of the machine while in no way impairing its portability or hardness. If you want to take your Model 100 on camping trips, use it on construction sites, or enter livestock data from horseback, this is the option for you. It entirely preserves the original character of the machine, except that it is not cheap. The big disappointment in the release of the Model 102 is that it is no easier to add memory above 32k.

Radio Shack has a portable, battery driven disk drive for the Model 100 for about \$200. It uses Macintosh style discs, holds 100k per disc, connects to the serial port, and has rather primitive operating software. It does involve getting carrying another piece of equipment, care of your disks and set up, so there is some compromise, but if you need to keep a lot of files with you on the road it's a godsend, and for a disc drive it's simple, durable and cheap.

Radio Shack has a good shoulder bag with enough room for a Model 100, a disk drive, and a portable printer; if you travel in more informal circles a back-pack works quite nicely. If you want to take notes in meetings or libraries you may need one other accessory: a set of orthopedic rubber bands to stop the clickety-clack of the keys. By carefully rocking the keys straight up and off you can place one or two bands on the key stem, then replace the key. This will almost completely silence the Model 100 proper; I haven't tested it on the 102. Do each key in turn.



Key removal tool: fabricate
from large paper clip,
use diagonally across key

Other Applications for the Model 100

All the major character based applications are available for the Model 100, as well as some additional unique applications that take advantage of its particular characteristics. The Model 100 is like the Macintosh in that it has a particularly enthusiastic user community, and because all Model 100 owners have modems that community is particularly active on-line. Extensive program listings are available. Applications available for free on-line or as BASIC listings in back issues of Portable 100 magazine include full fledged spreadsheets and idea processors and enhancements to Text and Telecom. In the Bay area the Well has a valuable sig (go laptop) and Abacus On-Line (415) 939-1246 is a BBS especially oriented to Model 100 callers—no menu over 8 lines. CompuServe has a very active Model 100 group.

If you are going to make intensive use of additional programs, the preferred format is a on a ROM chip, and the preferred ROM chip is the Super ROM from the PCSG. A program on a ROM chip is



easily inserted into the extra ROM socket, leaves the whole of the RAM space for data, and is instantly available on the main menu when you start up. Super ROM includes four programs on one chip for \$199: Write ROM, a word processing enhancement that adds on to Text; Thought, an outliner that has Think Tank compatible files; Lucid, a spreadsheet; and Lucid Data, a relational database. Write ROM has a Show Page function for printing, will let you set up prompts (as for an interview questionnaire), and can be configured to support any printer, including serial printers. It will therefore print with the Imagewriter, a lifesaver when your Macintosh is down, and is the only one of the integrated ROM chips to do so. Lucid is very fast, has a host of features, uses very little memory for a given size of spreadsheet, and can save or load data as text files. It got four out of four discs from InfoWorld, Byte said it was comparable to Lotus 1-2-3, and Model 100 reviewers were ecstatic. Lucid has the kind of reputation that makes you itch to try it out.

There are some fairly unique or specialized uses for the Model 100. It is used to control feed mixes or oil wells using an input-output board, as a brain for mobile testing instruments, or as a terminal with a portable phone. Since Radio Shack is inhospitable to third party products, the primary channel of information about specialized uses is Portable 100/200 magazine; the primary marketing channel for enhancements is mail order. Portable 100 is about one-third mail order catalogue, one third hacker's and user's newsletter, and one third ordinary computer magazine: if you are going to go beyond the basic functions of the Model 100 it is an essential resource, especially if you don't use the on-line services. (Occasionally available on newsstands, or somewhat more reliably for \$24/year from Portable 100/200, P.O. Box 250, Camden, Maine, 04843.)

How Much Will It Cost?

The Model 102 now lists for \$499 at Radio Shack. The Model 100 was discontinued this summer but is available used; it listed for 499 but was often on sale for \$400. The 102 actually costs less to make than the 100 so we can expect sales or price decreases, perhaps in the New Year.

Cost to Set Up The Model 100 for Use With The Macintosh:

24k Model 102, with manual and soft cover:	\$500
MockPackage:	\$40
4AA alkaline batteries (good for 20 hours):	\$4
Imagewriter I type cable:	\$30
A/B switch:	\$35
Total:	\$610

Batteries can get expensive; buy extra batteries on sale and keep them in the refrigerator where they will stay fresh indefinitely.

If you are a student or are otherwise constrained for money you can set up for considerably less. Authorized Radio Shack mail order dealers sell Tandy equipment (with warranty) for about 25% off (Happy Hands ((800) 545-9019 has a good reputation). The one pound (20%) weight reduction of the model 102 is of real value in a computer you are carrying around every single day, but if you are willing to sacrifice that, you can buy a used Model 100. There is little to go wrong and service immediately available. Create a file, turn the Model 100 off and on to see that it is still there; enter, and if possible try out Telecom and Term Mode in Telecom. With all files deleted, "memory available" on the main menu will be just less than the rated memory of 8, 16, 24, or 32k. You should be able to get a used 24k Model 100 for \$250 (Radio Shack's final clearance price), or less.



The Model 100/102 gives you all the freedom of a portable for an affordable price. It lets you go ahead and get the benefits of a portable now, without making a heavy investment in an area where the technology is still advancing rapidly. No other peripheral so far extends the range of your Macintosh. The rapidity and ease of work at the Macintosh stretches out to your whole working life; and the Macintosh incorporates the Model 100 within its desktop environment, letting you edit, cut or paste as easily as if Text were but another program written for the Macintosh.

Something Extra

For someone who has learned about computers from the Macintosh, the no-frills quality of the Model 100 is one of the distinctive characteristics of using it. On one hand there are no dialog boxes or MiniFinders, no fonts or rulers: just the simplest and most essential elements to enter text. On the other hand, you are so elementally close to the hardware that you can actually change the way things work with a sentence or two in BASIC.

Since there is no MiniFinder, and file names only six letters long tend to be cryptic and hard to remember, it is sometimes tough to enter the exact file name in Telecom. After you've run this program, you'll be able to press F7 before you upload and get a listing of your files within Term mode. F7 will give you the amount of memory free. Enter BASIC from the main menu and type in New <enter>. Type in the program.

```
10 FOR i = -1268 TO -1265 <enter>
20 READ X : POKE i,X <enter>
30 NEXT <enter>
40 DATA 172,126,58,31 <enter>
```

Proofread each line carefully before you press Enter. Be sure you have a space after each line number; otherwise they are not critical. Then press F4 "Run." If you get an error, type Edit <enter> which will put you into Text. Correct the line with the error until it is exactly as above, then press F8 to exit back to BASIC itself, and repeat F4 Run. Once you are successful go to Telecom, put it in Term mode, and try out the new function keys.

Gordon Haig is a sheet metal worker who uses his Model 100 to write articles like this while commuting on BART. He can be reached at (415) 549-2252 or through BMUG.



A Day at the Races with MacCharlie

Vicken Khachadourian

I was probably the last person who would have anything to say in a Macintosh user's newsletter. I have been a faithful IBM PC user, and icons on the Mac were ancient Greek signs until a special request from Apple Computer made me a happy user of the easy-to-understand System Folder.

On October 19, 1986, Apple sponsored a fun-run called "Silicon Rally". The course was 12K (that's kilometers, not kilobytes) for adults, plus some shorter Little Feet Heats for kids, through the city of Cupertino. Post-race activities included food and music and a product open-house -- mainly for the Apple IIGS -- in the beautiful Apple Fitness Center.

As the score keeper of many races, I was asked to tabulate the results of Silicon Rally with the condition that the Mac be used at the finish line to gather data. To do this, Apple supplied me with two Macs and two MacCharlies.

The MacCharlie is a box that conforms exactly to the right side of the Mac so that together they look like one machine. MacCharlie has its own 8088 microprocessor and the Mac is converted to a terminal by the start-up software. MacCharlie has two disk drives that load MS-DOS into its RAM. With a prosthetic keyboard that fits around the Mac keyboard, you get all the usual PC function keys. The system starts with the Macintosh operating system, and the user opens the MacCharlie icon that appears in the Finder. A menu bar appears which you use to run programs from the IBM disk. Another powerful and very convenient feature is ASCII file transfer from the MacCharlie disk to the internal drive of the Macintosh.

Scoring a race begins in the weeks before the event when information from entry forms, including an assigned race number and bar-coded age and sex categories, is put into a relational data base written in dBase II. On race day, the bib number will be used to retrieve the rest of the information on the person.

I score races with a program called "Your Exact Time" which I wrote for the IBM in Microsoft Pascal and Assembly. YET assigns a function key to stamp the PC clock next to incoming runner numbers, as spot-checked by a single operator at the finish line. For Silicon Rally, collection of times was done on the MacCharlies running on generators.

Further down the finish line, a bar-coded tab is ripped from the bib number as the runners file through the chutes, and the code is read into the computer with a light pen. Finishing time is then assigned by the order of the bar-coded tabs, making use of the spot checks. Macros in the data base program calculate each runner's overall place, and division place by age and sex. These results must be ready in time to announce the winners for the awards ceremony shortly after the race.

The concept may look pretty simple, but the difficulty and suspense of the whole process is that you are supposed to perform the described tasks in a period of thirty minutes in the midst of a noisy, sweaty, milling crowd. In any process where the seconds count, data collection is a definite bottleneck. Timing finishers requires as much coordination as playing a video game, especially in the swell of runners around "middle of the pack" (median) time.



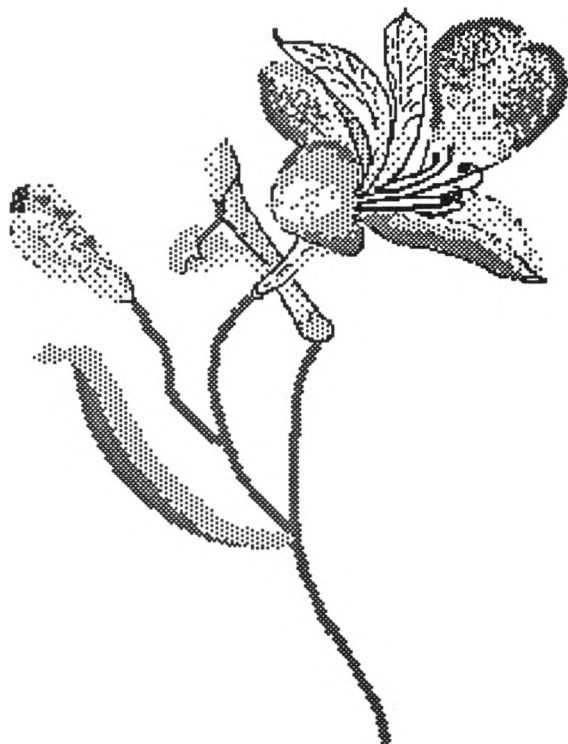
I have some things that I would like to see added to the MacCharlie and the Macintosh. Neither has expansion slots, and for the custom programmer, that is an absolute necessity. Another inconvenience was the propagation delay due to the fact that the Mac is a terminal to the MacCharlie. The elapsed time between a keystroke and the computer's ready signal was longer than usual, and if you are timing races, that may make a significant difference. Otherwise the delay is negligible.

The manufacturer, Dayna Communications, was very quick in providing me with technical and moral support. Converting my programs to make them run on the Mac seemed troublesome at first, but MacCharlie helped me solve that.

Gathering the race times and saving them to disk ran perfectly without a hitch, concluding my brief relationship with the Macintosh and MacCharlie. I am back to using the PC for my races, but having the powers of both the Macintosh and the PC in one machine has got me seriously thinking about the advantages of such a set-up for a Macintosh user.

MacCharlie \$1195 and \$1895
Dayna Communications
50 South Main Street Suite 530
Salt Lake City, UT 800-531-0600

Vicken Khachadourian is a 1986 graduate in Math/Computer Science at U.C. Berkeley. During his last year of school, he developed race registration and scoring software and started a small software company called "The Final Result". His race credits include the SuperRun II America covered live by ABC's Wide of Sports.



Tiger Lilly/Chan

BMUG
Fall 1986



(C) COPYRIGHT 1985,1986 ALL RIGHTS RESERVED
THE FINAL RESULT, INC.

#1

SILICON RALLY 1986 FOR OFFICIAL USE ONLY
RACEDAY APPLICATION FORM

LAST NAME: WOZNIAK FIRST NAME: STEVE
ADDRESS: [REDACTED] SEX: M AGE: 36
CITY: [REDACTED] STATE: CA ZIP: [REDACTED]

WAIVER - PLEASE NOTE THAT THIS IS A LEGAL DOCUMENT. READ IT CAREFULLY BEFORE YOU SIGN. I know that running a road race is a potentially hazardous activity. I should not enter and run unless I am medically able and properly trained. I also know that there will be traffic on the course route. I assume the risk of running in traffic. I also assume any and all other risks associated with running this event, including but not limited to falls, contact with other participants, the effects of weather (including high heat, smog, and/or humidity) and the condition of the roads. All such risks are known and appreciated by me. Knowing these facts and in consideration of your accepting my entry, I hereby for myself, my heirs, executor, administrators, or anyone else who might claim on my behalf, covenant not to sue, and waive, release, and discharge Apple Computer, Inc.; City of Cupertino; Crippled Children's Society of Santa Clara, Inc.; Race Officials; Volunteers; and any and all sponsors, including their agents, employees, assigns or anyone acting for or on their behalf, for any and all claims or liability for death, personal injury, or property damage of any kind or nature whatsoever arising out of, or in the course of, my participation in this event. This release or waiver extends to all claims of any kind or nature whatsoever, foreseen or unforeseen, known or unknown. The undersigned further grants full permission to Apple Computer, Inc. and the City of Cupertino, and/or agents authorized by them, to use any photographs, videotapes, motion pictures, recordings, or any other record of this event for any purpose. Applications for minors will be accepted only with a parent's signature.

YOUR SIGNATURE: X [Signature] DATE: 10/18/86

SIGNATURE OF PARENT IF UNDER 18: X _____

***** PLEASE CHECK THE BOX MATCHING YOUR AGE DIVISION ON RACE DAY *****

		MEN			
☐	<----(7 _ 13)	M7 	☐	<----(40 _ 49)	M40
☐	<----(14 _ 19)	M14 	☐	<----(50 _ 59)	M50
☐	<----(20 _ 29)	M20 	☐	<----(60 _ 69)	M60
☒	<----(30 _ 39)	M30 	☐	<----(70 & over)	M70

-----CHECK APPROPRIATE BOX (SHORT RUN / 12K)-----

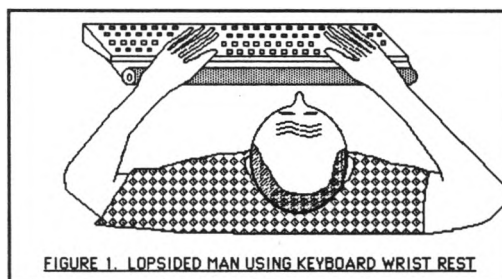
☐	<----(SHORT RUN)	H 	☒	<----(12K)	F
-----------------------	------------------	-------	----------------------------------	--------------	-------

A Day At The Races With MacCharlie
Example Silicon Rally 1986 Application Form

Keyboard Wrist Rest

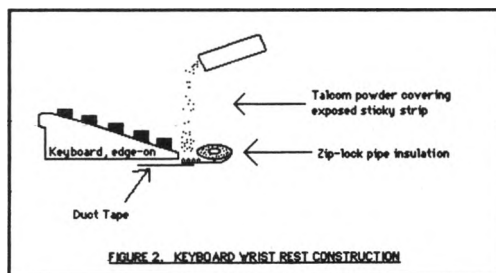
Kenneth William Hirsch

A piece of "zip-sleeve" pipe *insulation* makes a fine wrist rest for Mac keyboards. (See Figure 1.) I use the size intended for 1/2" copper pipe, from Ace Hardware (manufactured for Ace Hardware Corp. by Voltek, Inc. Lawrence, MA). There is also a larger size, for 3/4" copper pipe, which may be preferred by some people. Avoid getting other brands, which tend to be too thick and too stiff.



Zip-sleeve pipe insulation is made of closed cell polyethylene foam. It is ideal for firm, resilient support without absorbing sweat, coffee spills and other detritus of creativity. In addition, it is cheap. Four 36" tubes (enough for 8 Mac+ keyboards) cost me \$6.77 at my local Ace hardware store. Possibly one can get it elsewhere, but keep in mind that it was "Manufactured for Ace Hardware Corp. by Voltek, Inc...".

This could be another Mac users group "Make Your Own" night. The hardware needed includes: scissors, duct tape, talcum powder and the foam cut to size. Use one 3" strip or two 2" overlapped strips of duct tape. Set the foam close to the one end of the tape (tacky side up). Then, stick the other end of tape to the underside of the keyboard. Use talcum powder on any remaining exposed tape. It prevents the tape from wrinkling or crawling up to the edge of the keyboard.



A glance at the figure will show, there are three dimensions available for individual variation.

The idea for this keyboard wrist rest arose from use of an anti-chafe pad on an arm sling. After setting my dislocated shoulder, my bone doctor threaded the narrow nylon strap of the sling through a white foam tube. As to how I happened to be diving from a Maui cliff into the ocean, well that is another story...

BMUG Fall 1986



I wasted a lot of time phoning surgical supply houses trying to locate larger lengths of the tube to use for the wrist rest. Then my neighbor saw what I was calling "surgical cushion tube" and told me where babies came from. I wonder if Doc got the original piece of tubing from his plumber or from a passing stork. Hmmm.

Msg. #4105 in **Public Messages** Posted on 10/28/86 at 12:11:57

To: ALL From: BARBARA CHAN

Subject: BMUG DESIGN & GRAPHICS SIG

NOTES FROM OCTOBER 27, 1986 DESIGN MEETING

The Design group had Show and Tell last night with many members bringing artwork they had created on the Macintosh or Apple computers. The work ranged from animation to scanned illustrations, MacDraw artwork printed on the L300, and books created with JustText. Reid Knopp dropped by to show his MacArt portfolio of logos and trademarks which he creates in FullPaint, MacDraw or Macpaint and then animates with VideoWorks and sound. It was a lively demonstration and members offered suggestions to enhance the show. Reid uses this demo to show prospective clients his work. Carol Morita who is a freelance artist and Genentech's Art Director, showed the first six images she created for a company 1987 calendar. The images were not only fanciful but related to the various kinds of products and services that the biotech company performs. Her work was created in MacDraw and printed on the L300. Dan Wood demonstrated GraphicWorks and gave his opinions about ComickWorks, GraphicWorks and SuperPaint. The Works programs are object-oriented Paint programs. Arthur Luerhmann showed some of his beautiful PostScript artwork. All BMUG members will see them in the Fall newsletter. Arthur and his group write and produce books about the Apple line. They are currently working on an Appleworks book, using the Apple III UCSD Pascal editor and PostScript to create the text and screens. They used FONTographer to create an Apple II font. Barbara Chan showed a book she is editing and designing for UCSF--written and laid out with JustText, a heavy duty word processing and typesetting program from Knowledge Engineering. It gives you true typesetting capabilities and because it writes directly to PostScript, you can create interesting effects on the page. Kim and Robin who work at LaserWrite in Palo Alto showed some scanned artwork as well as fine artwork which combined scanned images and traditional silk screening.

Barbara asked Kim to create some autumn leaves for the Fall newsletter header. Longtime member Jerry Grigsy who works at the Oakland Tribune showed how the Oakland Trib is pasted up by hand with traditional wax methods. He also talked about the upcoming paper strike by typesetters who fear that computers will take over their jobs. Jerry has been creative enough to move into the computer field and worked with an author on laying out a science textbook with MS Word and MacDraw this summer.

The group was asked to think about design and layout ideas for the Spring newsletter with the goal of having the Design group do the design and production next time. People signed up to work at the Mactoberfest on November 4 from 11 to 8 at Pauley Ballroom. We need people to demonstrate word processors, graphics programs, animation programs, and scanners.



Locking your Macintosh

Cliff Stoll

Your Macintosh is easy to move -- all too easy to borrow. Several things are available to hold the Mac in one place; if your Mac is in a public place or your apartment's likely to be ripped off, you might consider one of these security kits.

Apple's security kit for the Macintosh [part number M-0201] has a four foot long cable that you wrap around to something solid. The cable connects to your Mac using steel inserts which plug into holes on the back of the Mac the keyboard. The mouse and external disks are protected by substituting Torx screws for the thumbscrews on their connector blocks.

If you're worried about reducing the chances of your Mac evaporating, Apple's security kit might be adequate. There's no interference with using your Mac; you can still move your Mac around your desk. Those steel inserts are a minor annoyance, since they stick out of the Mac when carrying it around. It's pretty cheap: about \$30 for the kit and another \$10 for a padlock.

Unfortunately, Apple's security kit just isn't very secure. The 1/8 inch steel cable can be cut; the Mac's case can be opened up and the steel insert removed; or the plastic case can be mangled and the steel insert removed. Since the plastic case is so easily broken, the cable kit will only delay a determined thief.

The Tosh is a Macintosh holddown device which encases the bottom of the Macintosh in a metal cabinet. By routing cables from the mouse and external disks through the cabinet, peripherals are protected somewhat. It's quite a bit more secure than a cable system since it wraps around the case of the Macintosh, preventing the cabinet from being opened.

You can't easily move your Macintosh around when it has a Tosh attached because the metal cabinet gets locked to your table top. Some of the cooling holes are covered up, and you'll curse when you want to get to some of the Mac's connectors or adjust the screen brightness control.

Certainly, the Tosh is more secure than the cable security kit from Apple. It costs around \$110, and well worth it if your Macintosh sits unwatched in a public area. Contact Doss Industries at 1224 Mariposa, San Francisco, 94107.

There are other possibilities for the paranoid. A steel cable can be locked through the chassis so that the printed circuit board would be broken if the Mac was ripped off. LaserWriters can be tied down using security kits designed for IBM PCs. If you have an alarm system, you might consider looping a wire through your Mac to set off the alarm when broken.

Msg. #1974 in **Public Messages** Posted on 07/03/86 at 23:57:33

To: ALL From: MICHAEL LAMOUREUX

Subject: Micah hard disk

I'm using a Micah internal hard disk with my Mac+ that I like a lot, though sometimes I would like to be able to shut off the drive, and use the Mac alone in peace and quiet. Does anyone know whether I can add a on/off switch to the drive and still have the Mac work properly?

Also, does anyone know the phone number for Micah's support BBS?

M.Lamoureux



Hush, Micah!

Michael Lamoureux

The Micah 20 Meg internal hard disk is a great drive...except it doesn't have an "off" switch. When I first got my Micah, I didn't mind the slight noise it made, but after a while I became envious of those owners of external hard drives who always had the option of turning off their disk whenever they didn't need it, or didn't like the sound...or who wanted to try out the latest alpha-test of some program that was sure to crash their hard disk. A quick call to Micah technical support confirmed my suspicion that turning off a drive couldn't hurt it, so out came the tools and now I have an on/off switch for my drive. Here's how to do it.

Electrically, all that is done is to insert a switch in the power line that runs between the on/off switch of the Mac and the Micah drive (see **Figure 1**). Use a switch that has a high enough rating for the power and proper cables. I used a switch rated for 3 amps at 125 volts AC, yet that was small enough to fit inside the hole for the security cable on the back of the Mac. In fact, I went one step further and instead of cutting the power line to Micah, I made two connectors that were wired together with the switch. Then one connector plugged into the Micah drive, and the other plugged into the power cable which had been unplugged from the Micah (See **Figure 2**). This has the advantage of being easily removed if you decide you don't like the idea. Use reasonable caution when you're doing this construction, since there are dangerous voltages present in the Mac, even when it is turned off.

Obviously, you shouldn't turn off your hard drive when it is in the middle of writing to the disk or something horrible might happen. In fact, you should not touch this new switch except when the Mac is turned off or is in the middle of the "beep" during a reboot. This is how I have used it for several months and I have never lost any data. Think of the switch as a "personality changer" for your Mac, which changes it from a hard disk machine to one without...and never change the personality of your Mac while it is running!

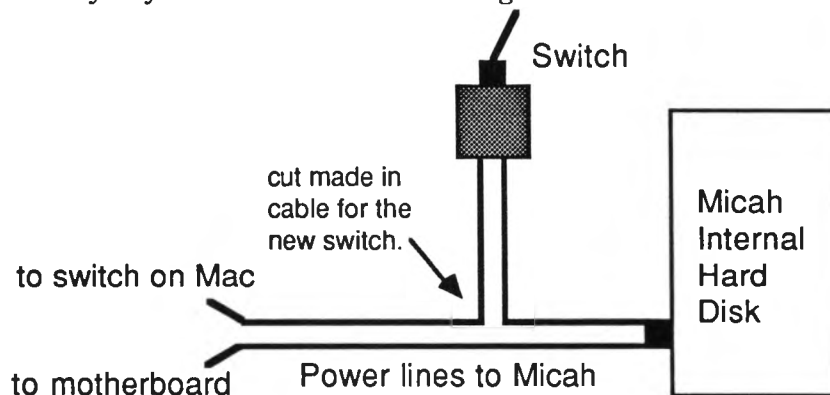


Figure 1

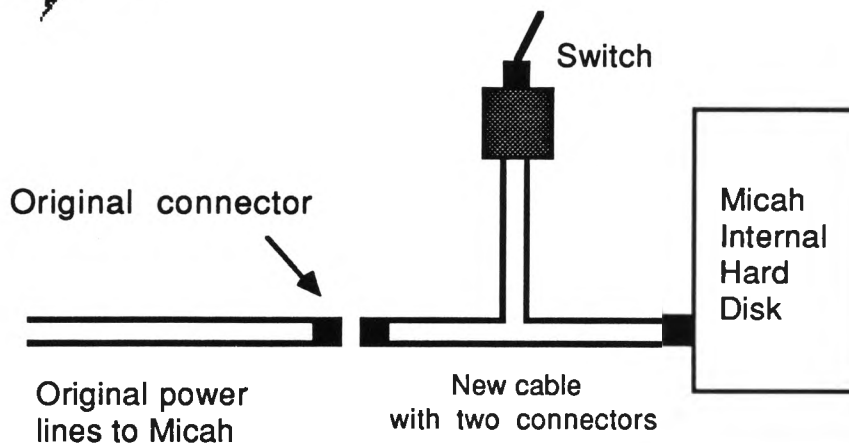


Figure 2

Msg. #2471 in **Public Messages** Posted on 08/08/86 at 20:26:54

To: TIM STANDING From: JAMES ULLREY

Subject: Reply To 'Mac Hard Disks'

HELLO TIM, THE SOFTWARE TO DRIVE THE HARD DISK IS, YES, THE MOST COMPLICATED PART. I HAVE A CONTROLLER CARD FROM WESTERN DIGITAL FOR ANOTHER APPLICATION AND THE MANUAL HAS A SECTION ON PROGRAMMING. I HAVE NOT STUDIED IT YET BUT I EXPECT IT TO REVEAL SECRETS OF THE TASK. I ALSO HAVE AN ARTICLE FROM BYTE MAG OF SEVERAL YEARS BACK CONCERNING ADDING A HARD DISK TO A S-100 Z-80 SYSTEM, AND LIKE WISE IT CONTAINS INFORMATION. THE ARTICLE BY JOHN BASS IN DR. DOBBS JOURNAL CAN PROBABLY BE A STARTING PLACE, ALTHOUGH I UNDERSTAND THAT THE C CODE IN THAT ARTICLE HAS SOME BUGS.

THE DISK CONTROLLER THAT HAS MY INTEREST, AS I SAID, IS THE SHUGART CARD. I HAVE NOT YET A C COMPILER NOR ANY C PROGRAMMING EXPERIENCE BUT IT SEEMS TO ME THAT THE DRIVER FOR A HARD DISK IS A PROBLEM SIGNIFICANTLY INTERESTING TO MOTIVATE LEARNING TO PROGRAM IN C. 8-8-86: I SHALL BE OUT OF TOWN FOR THE WEEKEND. MY NUMBER AGAIN 222-4698. IF A COOPERATIVE PROGRAMMING EFFORT INTERESTS YOU I WILL STATE THAT I HAVE SUCCESSFULLY PARTICIPATED IN GROUP PROGRAMMING EFFORTS. REGARDS, JIM



Tips for Hard Disk Users

Dan Callaway

Almost immediately after getting my Mac Plus, I went out and bought a 20-meg SCSI drive to go along with it. I now fail to see why anyone, given the choice, would be without a hard disk. At the time of writing, SCSI drives cost only a little more than twice as much as external 800k drives, but have 25 times the storage capacity and 3 or 4 times the speed! I bought a RELAX TECHNOLOGY Hard 20 Plus. It's a bit on the noisy side, but it's been performing flawlessly for 4 months now

When I got home I loaded ALL of my software onto the drive. It just fit. However in using it I noticed that going to and from the Finder took as long as it did with floppies! The Desk Top had swelled to 90k! Every time I quit to the Finder, this enormous DeskTop had to be reloaded. I needed to do some paring down. When was the last time I used MS BASIC? or played MacCheckers? I made a list of the applications I really used, and moved everything else onto floppies. The 800k disks are reasonably fast, so why keep these seldom-if-ever used programs on the hard disk? I now have about 10 megs worth on the drive, and my system runs much faster.

Backups

Everyone knows the importance of backups - especially with a hard drive. The first additional piece of software I bought was an HFS backup utility. It looked great at first. However as time went on I noticed that each incremental backup required more disks, though the files on my hard disk took up roughly the same space. The backup utility simply copied every file that had been changed since I last did a backup. For example, if I opened up my OverVUE files and made a few changes, the new file would be backed up without deleting the old one. With this kind of back-up, if you do any programming, you'll soon find your backup disks filled with several copies of the same source file! Or if you get a new version of a program, why keep back-ups of the old version? (I don't know if all backup utilities work this way, but I suspect they do)

I have found it not only more efficient but far easier to keep my files backed-up using the Finder. View your documents by Date, and selectively copy to disk only those files that you want to back-up. It may take a little longer to restore your data in the event of a crash, but it sure simplifies your work-a-day backups. Plus your data will be stored in normal Mac format, un-compressed, which gives you a little extra measure of security. You don't have to depend on the reliability of your restore software.

Fine tuning the System

One nice thing about a hard disk is having just one System and Finder to deal with. You can afford the time and the space to create a fully customized set up.

One Desk Accessory every hard disk user should have is Darin Adler's marvelous SkipFinder DA. Once you have activated SkipFinder, quitting an application will present you with a standard HFS GetFile window, allowing you to launch another application (or go to the Finder) without touching the Mouse. The transfer is so rapid, I have all but abandoned Switcher. When choosing names for folders, try to use a different first letter for each folder on the same level. This will enable you to select a folder by typing just the first letter, thereby shortening your visits to the GetFile boxes significantly. If you have the "expanded" keyboard the cursor keys are invaluable aids in GetFile boxes. Once accustomed to your filing structure, you can whip through your deeds.

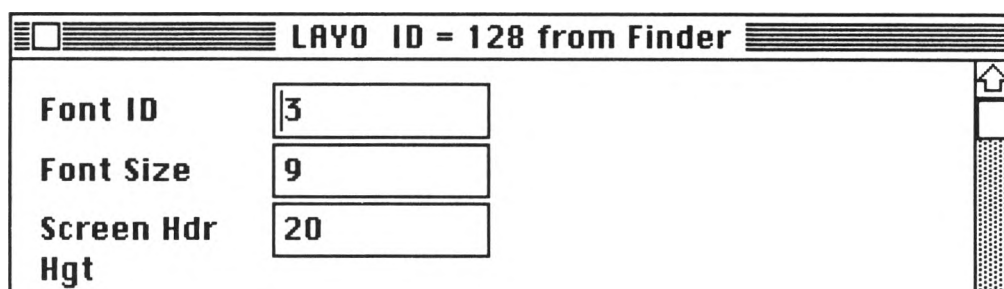
You can install up to 15 Desk Accessories. However you give yourself more flexibility if you leave one open slot for use with "Other..." or for application specific DA's. The "Other..." DA allows you to use uninstalled DA's; Desk Accessories that you find valuable but use infrequently remain available if kept on the disk as Font/DA Mover files. For those that you call frequently, but only in one application, use option/open in the Font/DA Mover to install the DA directly into that application.

There is a temptation to stack boatloads of fonts into your System file. If you're involved in graphic design, or own a LaserWriter Plus, this is fine, but I find searching through a long list of fonts a pain. 98% of the time I use one or another of three fonts. Who needs Venice font in an editor? You can avoid menu-crowding by installing special fonts directly into an application. Use ResEdit. Or open an application from the Font/DA Mover by holding down the Option Key while clicking on OPEN. Option/Open allows you to open ANY file - not just System or Font/DA files - and install a font or DA into it.

If you're a fan of FKEY's, as I am, you may want to write down a list of the ones you have installed in your system. I have about 5 of them, and I can't honestly say I remember which one is where at the present. My favorite is Léo LaPorte's Sort Key, which does a super-fast sort of text lines in the clipboard.

Customizing the DeskTop

It's possible to modify the Finder (5.3) to display more information on the screen. Using ResEdit, open the LAYO resource in the Finder. Here you'll find the settings the Finder uses to display icons and text on the screen.



Font ID	3
Font Size	9
Screen Hdr Hgt	20

(for a complete description of this resource, see "Inside Improvements" -Mac User, Sept.1986)

If you tend to keep your file names short, you can align your file icons closer together by reducing the "grid" size in the *Icon Horz. Spacing* and *Icon Vert. Spacing* boxes. The default is 64 x 64, but I found 54 x 54 to be more suitable. The icons will automatically align themselves to this invisible grid if you have the *Always grid drags* button enabled. If you tend to have long file names, then you should instead try changing the *Icon Vert. phase* to 30. This will "stagger" the placement of icons on a clean up so that long file names will not overlap.

If you prefer text to icons, you can specify the default for a new window to be View by Name, Date, Size, or Kind. Set *Default view* to the number corresponding to the place your preference occupies in the Finder's "View" menu. For example, for the third menu item, by Name, set *Default view* to 3. *Line spacing* specifies the number of pixels between lines in textual Views. If

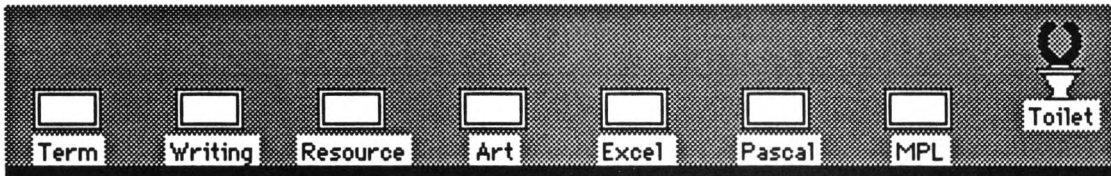


you change this value to 12, you'll be able to display 25% more files than with the standard 16-pixel spacing. In considering this, keep in mind that I have very good eyesight!

The first 5 *Tab stops* specify the distance between Name, Size, Kind, date last modified, and time last modified. Normally your window would have to be almost as wide as the screen to see all this at once, but by reducing the Tab stop values, you can make the whole display visible in a smaller area. I have my Tab stops set at 20, 95, 135, 190, 245. This shows me all I need to know in about half the space. I've also changed *Text view date* to \$0000, which gives me the date as MM/DD/YY instead of the overly verbose way Apple decided to display it.

Disabling *Use zoom Rects* will speed up the Finder by a fair bit. Apple apparently preferred a pleasing visual interface over raw speed. I'll take the speed. Whereas you disable *Use zoom Rects* to eliminate the zoom rects, you enable *Skip trash warnings* to eliminate the trash warnings. If you hesitate to do this, just ask yourself how many times you've had to "cancel" a file deletion. (I wish some DA's could have trash warnings nixed)

While you have the Finder open in ResEdit, you may want to shrink the size of the folder icon. I found it too cumbersome, especially since it is by far the most common icon on the screen. Just open the ICN# resource, double click on the picture of the folder, and slice away in the Fatbits display. I tend to modify my icons from time to time, just as some people change the wallpaper in their bathroom.



You may want to keep your most-used folders permanently on the desktop. Otherwise you must open the hard-disk icon, then the folder, each time you want to open a file. Besides, it gives the desk top that "lived in" feeling.

One last word: Remember, when modifying programs with the Resource Editor, be sure to throw out every other copy of that program. It's important that you have no back ups, for they will only confuse you in the future...

Dan Callaway
One Tisbury Lane
Richmond CA 94805
237-3548h
526-4444w

Building Your Own Hard Disk

Tim Standing

At the beginning of the summer, I decided to build my own Small Computer System Interface (SCSI) hard disk. I spent a month learning all I could about what made a hard disk fast and reliable. Once I ordered the parts and assembled it, it took me an additional three months of writing and debugging the software to make it work. I am writing this article so that you can learn from my experience and build your own hard disk. In the following pages, you'll find all the information that you will need to buy the parts, assemble and format a SCSI hard disk.

Since I knew next to nothing about electronics, the idea of building my own hard disk seemed foolhardy at first. However, I do know quite a bit about programming and thought that I could write the necessary software. I had installed IBM PC hard disks and knew there was nothing to it but correctly connecting up the cables. I figured assembling a hard disk for my Mac was probably just as easy and that I could put one together without understanding how any of those funny looking chips worked. If you've made a few cables, you too should be able to make a hard disk.

Expect to spend about about 6 hours assembling all the parts. My formatting program, *U2 SCSI Disk*, can be used by anybody. It's easy to use and includes on-line help. With my software, it should take no more than 30 minutes to format a hard disk.

This article is divided into three parts; what you need to know about controller boards and disk drives, step-by-step instructions on assembly including a parts and suppliers list, and how to use my formatting program.

I was inspired to write this article after telling a friend, Dan Chin, how to put together his own hard disk. Without his encouragement and the enthusiasm of another friend, Bill Bargagliotti, I would never have spent all the hours in front of the Mac that were necessary to write this down. I am grateful to Virginia Rath who read various drafts of this article and made many helpful suggestions. Thank you very much, Dan, Bill and Virginia (all at UCSF).

Understanding Hard Disks

When Apple came out with the Mac Plus last year, they included a SCSI port to allow people to attach hard disks. Manufacturers can build SCSI hard disks that are fast, cheap, and reliable. The SCSI port also allows users to attach as many as seven peripherals (hard disks or tape drives) to their Mac.

The SCSI interface is a standard way of connecting microcomputers with peripherals (such as hard disks). The connection between the microcomputer and the peripheral is made using a parallel interface. A parallel interface sends each data byte (eight bits) down eight wires simultaneously (one wire for each bit). This interface is much faster than a serial connection where all eight bits are sent down the same wire one after another. In addition, the SCSI connection uses a bus. This means that all the devices are connected one together in a line, with the Mac at one end. When the Mac wants to talk to a hard disk on the bus, it first sends the hard disk's number (called an address) down the bus to see if the hard disk is there. Each device on the bus must therefore have a unique address so that two devices won't respond to the same command.

Your SCSI hard disk will have three parts: a disk drive, a controller board and a box with a



power supply. The box and power supply protects the electronics and provides the necessary power. The disk drive stores information (files, programs, etc.) The controller board acts as an interpreter. It determines which of the Macs' SCSI commands are addressed to it and it translates these into commands that the disk drive understands. The controller board tells the disk drive where to read and write from, and it checks the data for errors. If it finds errors, it asks the disk drive to repeat the operation. Error-checking and correction occurs without the Mac knowing about it.

Choice of Disk Drive

I recommend that you buy a 5 1/4 inch disk drive from a manufacturer with a lot of experience. Seagate (see the parts list for more information) has a good track record. Although they don't have the lowest prices, they do make good disk drives and stand behind them. If you have problems with your disk drive (like it makes a whining noise when you start it up or doesn't start at all), double-check everything to make sure that it's hooked up correctly. If it still doesn't work, call the technical support folks at the place you purchased it and explain the problem. Even the best disk drive manufactures have a 1 - 2 % field failure rate (disk drives do get dropped during shipping).

All disk drives are not created equal. There are five things you should know about the disk drive you are thinking of buying.

1 Disk drives range in formatted size from twenty megabytes up to several hundred megabytes. The size you choose depends on your needs and how much you want to spend.

2 You need to know if the drive is certified for Run Length Limited (RLL) encoding. By reading and writing at a higher density, RLL encoding allows you to get 50% more data on a given disk drive. Not all disk drives can operate at this higher density. RLL encoding also requires a special controller board. If you use a RLL controller board with a non-RLL certified disk drive, it may not work. A RLL controller board and the Seagate ST 4051 disk drive (non-RLL certified) didn't work.

3 Does the disk drive automatically "park its' heads" when the power is turned off. "Parking the heads" means that the heads get locked in a special position when the disk drive is not being used so that they don't bounce around scratching the surface of the disk. This is especially important if you plan to carry your hard disk around a lot. Many disk drives require a specific command to park the heads. If the disk drive you use requires a specific command make sure the software you use includes it.

4 Find out how the disk drive moves the heads across the disk. There are two ways of doing it. One method is to use a stepper motor just like the slow floppy disk drives on your Mac. Disk drive speed is measured by its average access time (time required for the head to move 1/3 of the way across the disk). Drives with stepper motors have average access times between 65 and 90 milliseconds. The second method is to use a voice coil similar to the coil in a stereo speaker. A voice coil is much faster but also more expensive. Average access time for voice coil disk drives ranges from 25 - 35 milliseconds. Usually these drives automatically park the heads when you turn off the power.

Average access time will not affect the time it takes to read a file or start a program when the hard disk is new, but it will after you've been using it for a while. A relatively empty hard disk stores files in sectors (512 bytes each) that are next to each other. Reading hard disk files that are next to each other does not require the head to move much. After quite a bit of reading and

writing, files end up with each sector located on a different part of the disk. Now the head must move a lot to read the file. Any operation that requires you to read a small file won't be affected by average access time (e.g. starting an application or reading a MacWrite document). However, operations that read many files or read large files are heavily influenced by the average access time of the disk drive (compiling, using a database, returning to the **Finder**, or opening a folder).

5 Finally, find out how fast your disk drive will accept step pulses. A single step pulse tells the head to move one track over. The faster you can issue a step pulse, the faster the head will find the track that it is looking for. The interval between step pulses is called the step pulse timing. The step pulse timing has more of an effect for disk drives that use voice coils than for those that use stepper motors. When you purchase your disk drive, make sure that you find out what rate it will accept step pulses. The choices that you have using the **U2 SCSI Disk** program are 12 microseconds, 28 microseconds or 3 milliseconds.

Older disk drives need 3 milliseconds between each step pulse. This is because they cannot count step pulses while the head is moving and have to wait for the head to reach the new track before they can accept the next pulse. A few years ago, disk drives became available that accept step pulses at 35 microsecond intervals. Step pulses issued at this speed require the disk drive to continue accepting and remembering step pulses while the head is still moving. Each step pulse is stored in a buffer; this is called buffered seeking. IBM adopted a step pulse timing of 35 microseconds for its PC AT computers. This is considered slow by current standards. Most disk drives that you can buy today will accept step pulses at 12 microsecond intervals. This is true of all the new Seagate disk drives.

I recommend a ST225 (20 Megabyte) disk drive from Seagate or one of the ST4000 series (also from Seagate). I've seen the ST225 advertised from a mail order house for as little as \$329 (California Digital, 17700 Figueroa St., Carson, CA 90248; (213) 217-0500). The ST4000 series comes in several sizes ranging from 20 megabytes (\$500) up to 90 megabytes (\$1100) and is available from Mini Micro Supplies (see parts list for address and phone number). I don't know anything about 3 1/2 inch disk drives.

Choice of Controller Board

There are at least three controller boards on the market that will work on a SCSI bus. I chose one made by Adaptec (model number ACB-4000A). It can control one or two disk drives and has many features that make it reliable and easy to use. Adaptec also makes a RLL version of this board called the ACB-4070. I know that the software I have written works with the ACB-4000A controller board, but I don't know if it will work with controller boards from other manufacturers.

The Adaptec controller board allows you to use any interleave from 1 to 16 (if you don't know what interleave is, you should skip ahead and read the section entitled "What is Interleave"). The program **U2 SCSI Disk** performs best with an interleave of 3. I think I may be able to speed up the software so that you can use an interleave of 2, so keep your eyes open for an update.

Choice of Box and Power Supply

Most boxes advertised in the back pages of IBM PC magazines will work for this project. You need to decide whether you want room for one or two disk drives. I bought a one disk drive box. It's an ugly grey color and even looks like an IBM PC. I wrote "IBM sucks" across the front just so my Mac wouldn't think that it was about to be replaced.

Get a box with a built-in "switching" power supply as these are less prone to interference from



Get a box with a built-in "switching" power supply as these are less prone to interference from your refrigerator or your landlord's vacuum cleaner. Switching power supplies use a sophisticated high frequency circuit rather than large capacitors to regulate output voltage. They are smaller, lighter, and create less heat.

Formatting

There are two types of formatting that are needed before a SCSI hard disk can be recognized by the Mac, low level and high level formatting. They are both important and serve totally different functions.

Low level formatting prepares the disk drive so that the controller board can use it. Without low level formatting, the disk drive can't hold information. Any attempt to read or write data to the disk drive will result in the controller board returning a error message.

Low level formatting involves first checking the magnetic coating on the disk surfaces for defects and then writing information about these defects and about the disk drive's characteristics onto the disk drive itself. Defects on the disk surfaces are called bad blocks. These are places on the surface of the media which cannot hold a magnetic charge. Disk drive manufacturers often supply a list of these defects (called a bad block list) with the disk drive when you purchase it. In my experience the bad block lists that accompany disk drives are not always complete. It is important to check the disk drive for bad blocks and then to append this list to the manufacturer's. Once you've created a bad block list you can continue low level formatting. The controller board puts a bad block map on the disk drive which tells it not to use the sectors that contain these defects. During low level formatting, the controller board also writes information to the disk drive such as its size and interleave number. When you turn the hard disk on, the controller board reads this information into its memory which enables it to talk to the disk drive.

High level formatting puts information on the hard disk that permits it to be recognized by the Macintosh. Before high level formatting the disk drive can read and write information perfectly well, but the Mac doesn't know it's there. Formatting programs like Apple's *Generic SCSI Installer* or the public domain program *SF&J* perform some of the high level formatting functions.

There are four things that must be placed on the hard disk during high level formatting. The first two are the Drive Descriptor Map (DDM) and the Device Partition Map (DPM). These allow a given SCSI hard disk to be partitioned into many volumes or to be used by more than one type of computer. Next, the driver must be installed on the device. The driver is a little program that tells the Mac how to talk to the SCSI hard disk. It includes the read and write functions and a subroutine that is used when you first turn your Mac on. Some of the differences between the various hard disks available result from the the way in which their read and write functions are written. For me, creating a good driver was the hardest part of making a hard disk.

Although the Mac can talk to the hard disk now, it can't store files on it yet because the hard disk lacks a file system. The file system is installed next by writing two files to the hard disk. These files contain the file structure for Apple's Hierarchical File System (HFS). One is called the **Catalog** file and the other, the **Extent** file. I am not exactly sure what these files contain, but any true Mac Hack should be able to tell you.

Lastly, the boot blocks must be written to the hard disk. The boot blocks tell the Mac what file names to look for to provide the operating system functions. The other high level formatting

programs I mentioned do not complete this step. Instead, you are instructed to perform some incantation in the **Finder** to get the **Finder** to write the boot blocks.

What is Interleave

The interleave factor describes how many sectors on a track are skipped before reading the next one. This number is tuned to the speed of your software. If your software can't keep up with the disk drive operating with a given interleave, reading and writing will actually go slower. The smaller the number the faster you can read and write information. An interleave of 1 means that you read each sector in the order that it occurs on the track. An interleave of 2 means that you read one sector, skip one sector. An interleave of 3 means that you read one, skip 2, read one, skip 2, etc. By skipping two sectors, you give the Mac time to catch up with the incoming data.

In order to use a smaller interleave number you must have faster software. The driver that I have written works best using an interleave of 3. This limitation comes from the way that the SCSI manager was written by Apple. If I have the time this winter, I'd like to rewrite the SCSI manager to make transfers faster. Taking advantage of this will require reformatting your disk with a smaller interleave number.

Read Precompensation and Write Current Reduction

As the head moves towards the center of the disk, data are written closer together. The exact method of reading and writing has to be altered slightly to compensate for this higher density. The Read Precompensation number and the Write Current Reduction number determines the first cylinder which requires these different read and write signals. You will need these two numbers in order to correctly low level format your hard disk.

When you buy your disk drive, make sure that the you find out what values to use for Read Precompensation and the Write Current Reduction. For a Seagate ST225, the Read Precompensation is 300. The Write Current Reduction is calculated by the disk drive itself, so you should use a value of 0. For the Seagate ST4000 series of disk drives, both of these numbers are calculated by the disk drive, so you should set both of them to 0. If you buy a Seagate disk drive and don't know what values to use, you can call Seagate Technical Support at (408) 438-5333.

Static Electricity

The chips used on the disk drive and controller board are very sensitive to static electricity. These chips contain very small transistors that can be easily electricuted by small sparks. Therefore, both your controller board and disk drive will arrive in static shielding bags with huge warning labels all over them. If you are like me, you'll be totally intimidated by these labels and afraid to open the bags and take things out. Here's what you have to be careful with to avoid frying any of these chips.

Leave everything in its bag until you are ready to use it. This will protect the parts from all sources of static electricity. When you're ready, spread out some aluminum foil and place the parts on the foil. Touch *only* the sides of the controller board and the metal frame of the disk drive when handling them. Avoid touching the components or any metal on the controller board or the disk drive board (this includes the connectors). If you must touch the components or connectors to plug in a cable, you should first touch both hands to the foil and then to the metal frame of the disk drive. By doing this *immediately*, before you touch any static sensitive parts, you will discharge all the static electricity that has built up on your hands. If you have to service your hard disk after you've put it all together, you should observe the same precautions.



If you follow these simple rules, you shouldn't have any problems.

Termination

What is termination and what do you need to know about it? Termination is a method of reducing noise caused by signals bouncing off the ends of the wires. This type of noise is a problem on any high frequency data bus like a SCSI bus. It is called termination simply because it is something that is done to both ends of the bus. Termination is accomplished by taking resistors and placing them on all of the data wires. Two resistors are used for each of the eight data lines. One of these resistors goes between the data wire and ground, and the other goes between the data wire and + 5 volts.

Although Apple should include termination inside the Mac Plus, they don't. They tell you that only the device closest to the Mac and the device furthest away from the Mac should be terminated. If you only have one device (excluding the Mac), then you only need one terminator. If you have two or more devices you will need two terminators. Terminators can be bought from your local Apple dealer (Apple part number M2559).

Putting It Together

Time

As I said in the beginning, it took me 6 hours to put all the hardware together. Dan Chin told me that it only took him 3 hours, but I guess he's fast. You should probably plan on spending a day doing everything. If you have assembled everything correctly, it will take no more than 30 minutes to do the formatting.

Tools Needed

Tool:

Regular Screwdriver
Philips Screwdriver
Soldering Iron
Pliers For attaching connectors to ribbon cable.
File, Hacksaw, Drill

Volt Ohm Meter
Magic Marker
Heat Shrink Tubing

What you need it for:

Putting the parts together

There are only 12 wires that need soldering.

For adding new slots for connectors on the back panel of the box that you use.

To check that cables are made correctly.

Parts List

Quantity:	Item:	Description:
1	Adaptec, ACB-4000A	Adaptec controller board.
1	Seagate, ST 4051	Hard disk drive (see the above section for suggestions).
1	Box with power supply	
1	Apple, M2559	SCSI cable terminator.
1	Apple, M2556	SCSI System Cable. This connects your Mac to the hard disk.
1	DIP switch	Buy a three station DIP switch. You will mount this on the back panel and use it for setting the SCSI address of your disk drive.
8 feet	50 pin Ribbon cable	
1	AMP Female connector	This is attached to the ground on the frame of

- 1 Spade Connector
- 1 20 pin IDE connector
- 2 34 pin IDE connector
- 1 50 pin IDC connector
- 2 20 pin IDC connector

- 2 Amphenol, #57F-50

- 1 foot Heat shrink tubing
- 1 1 1/4 inch fuse holder
- 1 1 1/4 inch fuse

- 4 6 X 32 X 1/2 inch screws

- 4 6 X 32 X 3/4 inch screws

- 4 6 X 32 nuts
- 8 Splined lock washers (metal)
- 4 1/4 inch standoffs.

- 8 Nylon washers

the disk drive.
To attach the ground wire to the chassis ground.
Edge connector for the disk drive.
Edge connector for connecting the controller board to the disk drive.
Female pin connector to attach to the controller board.
Female pin connectors. One is used for the connection between the controller board and the disk drive. The other is used to connect the disk drive address lines to the DIP switch mounted on the back panel.
50 pin SCSI connector. These are connectors for the back panel and include latches.

For back panel for termination power.
500 milliamp, 250 volt fuse for termination power.
They are used for mounting the controller board to the frame of the disk drive.
For mounting the SCSI connectors to the back panel.
Also for mounting SCSI connectors.

Also used for mounting the controller board to the frame of the disk drive.
These are to prevent your screws or standoffs from grounding any of the components on the board.

Where to Get Parts

I purchased the controller board from Wyle Laboratories (408) 727-2500. Their price for the Adaptec ACB-4000A is \$139. They have good service and great prices.

The box / power supply and the Seagate disk drive (model # ST4051) were both purchased from Mini-Micro Supply Inc., 1977 O'Toole Ave. Suite B108, San Jose, CA 95131, (408) 435-1977. They charged me \$120 for the enclosure (item # E001-P) and \$650 for the disk drive. The disk drive may seem expensive until you realize it is 43 megabytes and has 40 millisecond average access time.

The two SCSI connectors for the back panel (Amphenol #57F-50) were ordered from Cetec (408) 748-8550 and were \$14 each.

All the other electrical supplies I purchased from Zack's Electronics at 1444 Market St. in San Francisco (415) 626-1444. You could probably go to any electronics supply house and get them or order them from Cetec when you order the other connectors. All of the parts listed above cost me just under \$50.

The Apple parts I bought from my local Apple dealer. The screws and nylon washers I bought from my local hardware store.

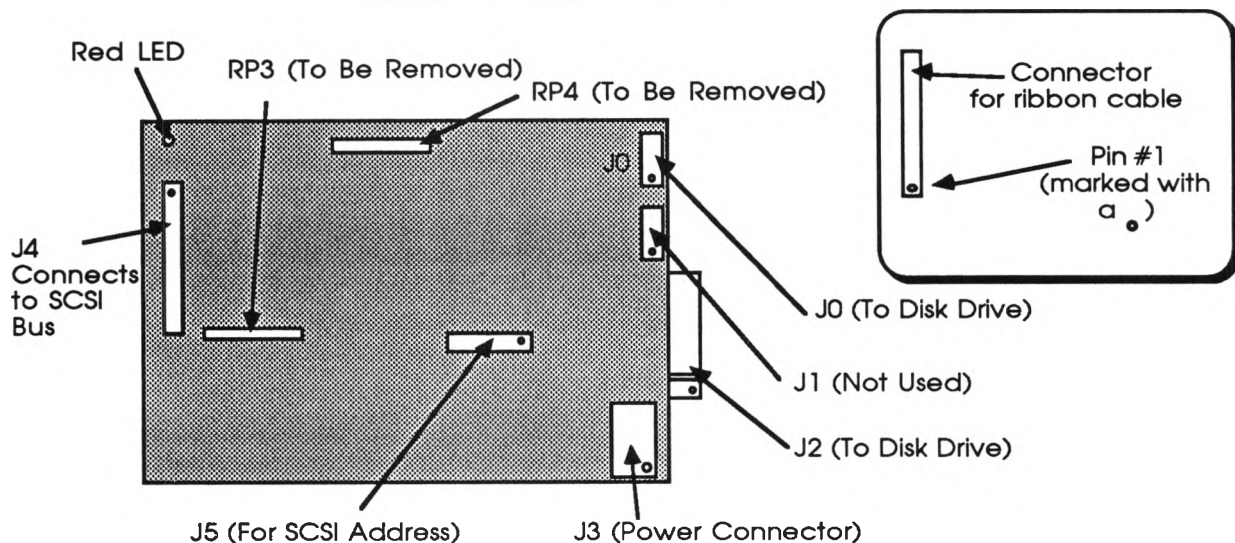


Before You Start

It's a good idea to wait and buy all the connectors and cables when you have already received the controller board and the disk drive. That way you can see exactly what types of connectors you need.

Examine the board and disk drive to make sure you can recognize all the connectors I've illustrated in the diagrams (see fig. #1). Notice that all the connectors are labelled as to which is pin #1. You will find the pin numbers on both male and female connectors molded into the plastic of the connector itself. It may take a while to find them as they are often less than 1/16 of an inch high. In addition, the pin numbers are written in white on the boards to which the connectors are attached. I have shown where pin #1 is for each connector in all my figures. In addition, the IDC and IDE connectors also have a triangle on the side nearest pin #1. I usually mark this with a black magic marker when I make up the cables so that I can easily keep track of pin #1 at each end of the cable.

Fig. 1 Adaptec ACB 4000A Controller Board
(Component Side)



None of the connectors used to connect to ribbon cable require soldering. They are pressed together with the cable between the two halves. You will need a pair of pliers or a small vice to apply enough pressure to get the two halves to mate properly. If you have never used this type of connector before, you should use a volt ohm meter to check that you've gotten good connections at both ends. If you make a mistake with one of the connectors, do not take it apart and reuse it. These connectors are not designed to be reused and will not make a good connection a second time.

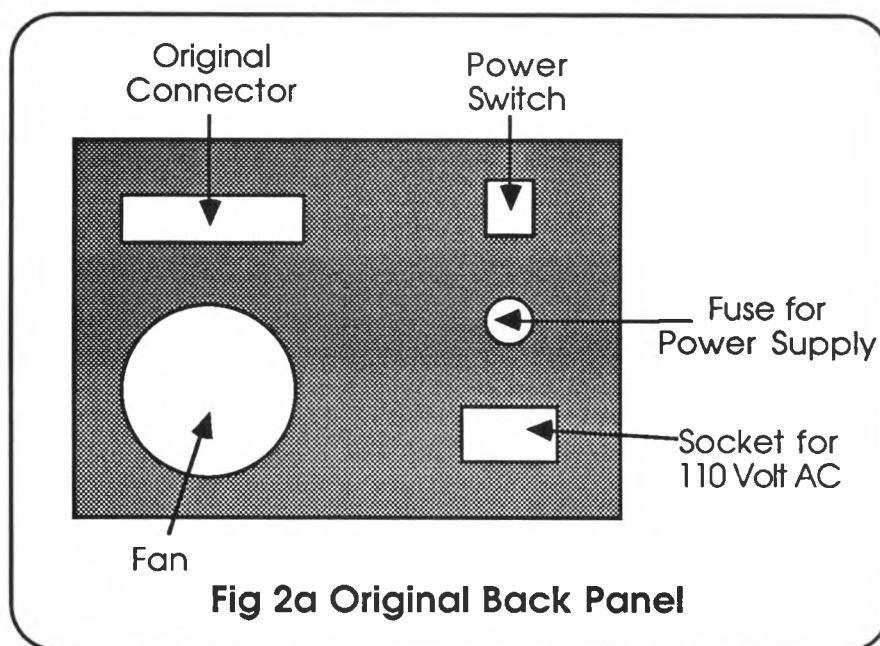
The ribbon cable has a red stripe down one side to help you attach connectors correctly. Always attach the connectors to the ribbon cable so that the red wire is pin #1. Assembling the cables carefully and checking them with a volt ohm meter will pay off in the long run. This is where 90% of my problems came from.

In addition to 50 wire ribbon cable, you will need 34 wire, 20 wire and 6 wire ribbon cable. Rather than buying four different sizes, you can take a length of the 50 wire ribbon cable and cut it between two wires with a razor blade to get the desired width you need. When you do this, be careful not to expose any of the wires by cutting through the insulation.

Step By Step Instructions

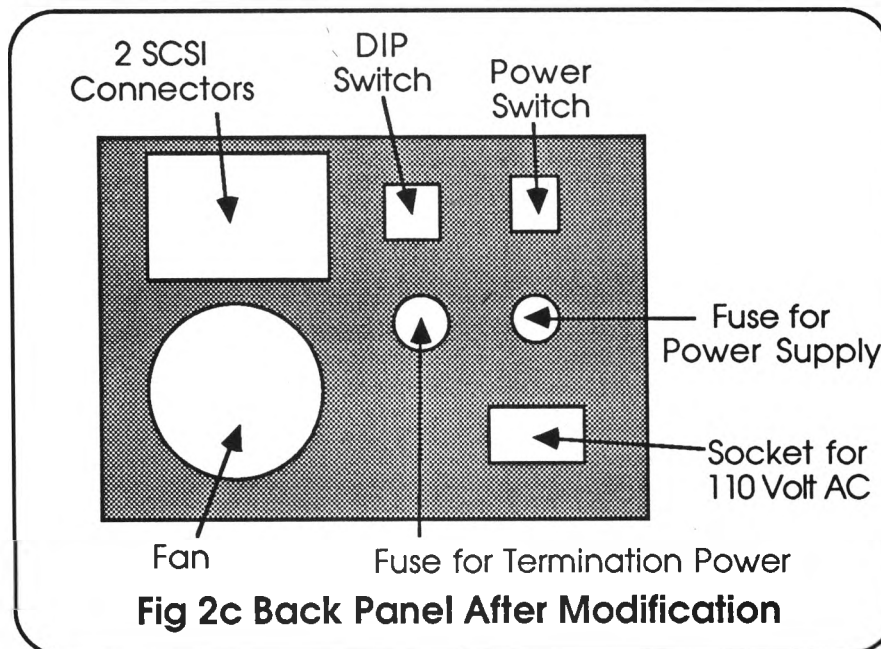
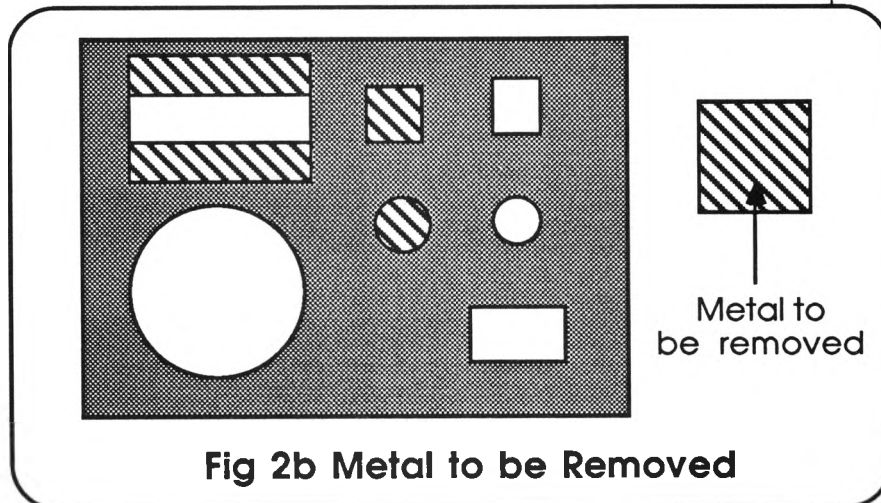
1 Remove the fan, power switch, power supply, fuse holder and all of the connectors from the box. Mark all wires and screws as you take them off so that you can be sure to replace them correctly.

2 Add several holes to the back panel of your box (see fig. #2). They will be used for mounting the two SCSI connectors, the extra fuse holder and the DIP switch. The fuse holder is used for the +5 volt power needed for terminating the SCSI bus. The three station DIP switch sets the SCSI address of your hard disk. You will set it as a three digit binary number using these switches. Don't set your disk to number 7 as this is the number that your Mac uses.



Start by making an opening for the two SCSI connectors. The center to center distance between these two connectors must be at least $\frac{3}{4}$ of an inch. This is because the cables that Apple sells for the SCSI bus all come with rubber hoods and require extra clearance. (Before you start, remember that you still have to be able to mount the fan on the back panel.) I used a hack saw and a couple of files to make this opening. You must also drill and shape holes for the extra fuse holder and for the DIP switch. Modifying the back panel is the hardest part of constructing a SCSI hard disk and will take you the most time. It took me about 3 hours to do.

3 Replace the power switch, power supply, fan, 110 volt AC socket and fuse for the 110 volt AC input power. Attach the extra fuse holder to the back panel. Make sure that both of the metal tabs point up so that you can more easily solder wires to them.



4 Remove the controller board from its protective bag. You will see two resistor packs that are mounted in sockets. They are labelled RP3 and RP4 on the board (I have labelled the same in fig. 1). They are usually red or yellow and have ten pins all in a straight line. Carefully remove both of these resistor packs using a pair of pliers. Be careful not to damage the controller board. You can throw both of the resistor packs away.

5 Mount the controller board on the frame of the disk drive. There are threaded holes on the frame of all 5 1/4 inch diameter drives for this purpose. (If your disk drive does not have mounting holes for the controller board, drill four holes in the base of the box and mount your controller board directly on the box.) The board should be mounted so that the component side faces away from the disk drive. This gives you room to mount connectors at J4 and J5 (fig. 1). In addition, the red LED on the controller board should be at the same end as the red LED on the



disk drive. This places the 20 pin and 34 pin connectors on both the controller board and the disk drive next to each other.

Use the 6 x 32 x 1/2 inch screws to mount the controller board to the disk drive. The 1/4 inch standoffs (those little metal cylinders) will be put between the controller board and the disk drive and will provide an air space for cooling. Put a nylon washer on either side of the controller board to insulate it from the screw head and the standoff. This prevents the controller board from shorting out.

6 Remove the mounting rails from the box if you haven't already done so. The mounting rails are the two pieces of sheet metal that are attached to the base of the box. They are designed to be attached to your disk drive, and will hold it in place. Attach the disk drive/controller board assembly to the mounting rails. There are screw holes in the frame of your disk drive for this purpose. When you mount the disk drive, the controller board should be facing the base of the box. There should be at least a 3/4 inch clearance between the controller board and the base of the box so that you can plug in the connectors at J4 and J5 (fig. 1). You may have to drill new holes in the mounting rails to provide this clearance.

7 Make a 6 inch piece of 34 wire ribbon cable from the 50 wire ribbon cable that you bought. Attach one 34 pin edge connector to each end. Make sure that pin #1 of the first connector is connected to pin #1 of the second connector. Attach one end to the 34 pin edge on the controller board at J2 and the other to the 34 pin edge of the disk drive. Make sure that the two connectors are correctly oriented with respect to the boards. Each male edge connector has a notch between pin #4 and pin #6. Use these notches to help you attach the connectors correctly.

8 Make a 6 inch piece of 20 wire ribbon cable. Attach one of the 20 pin female connectors to one end and the 20 pin edge connector to the other end. Check that pin #1 of the edge connector is connected to pin #1 of the female connector. Attach the 20 pin edge connector to the disk drive. Again make sure that you have it in the right orientation so that pin #1 is at the end which is notched. Join the 20 pin female connector to the 20 pin male connector on the controller board which is marked J0 (see fig. 1). The male connector is marked in white lettering on the controller board itself.

9 Now take a look at the remaining 20 pin male connector on the disk drive. It should have one jumper on it connecting a pair of pins which are next to each other. Make sure that it connects the pair of pins that are nearest the 20 pin edge connector. This sets the drive's address for the controller. The controller can only communicate with the disk drive when the disk drive is labelled with the lowest possible number.

10 Make a 2 foot long 6 wire ribbon cable. Connect one end of this cable to the last remaining 20 pin female connector. The six wires should be connected to the sockets numbered 1 - 6 on the female connector. Separate the six wires for 1/2 inch at the other end and strip 1/4 inch of insulation from each one. Solder wires 1 and 2 to the first switch on the DIP switch. Solder wires 3 and 4 to the second switch and the last two wires to the last switch. Now check that closing switch number one connects pins #1 and #2 on the female 20 pin connector. Close switch number two and check that it connects pins #3 and #4. The third switch should connect pins #5 and #6.

11 Epoxy the DIP switch into the hole that you cut for it in the back panel. Connect the 20 pin female connector to the controller board at the 16 pin male connector labeled J5 (see fig. 1). Make sure that the socket is installed with pin #1 of the female connector attached to pin A of



the male connector (labelled in white lettering on the board itself). You will notice that the male connector on the board has only 16 pins and that your female connector has 20 pins. This is okay, the extra 4 pins simply aren't used.

12 Cut a 2 foot piece of 50 wire ribbon cable. Attach the 50 pin female connector to one end. Connect one of the SCSI connectors to the other end. Again make sure that pin #1 from one connector is connected to pin #1 of the second. You will find that the SCSI connector is labelled with tiny numbers. These numbers are on the outside which is the side with the latches. The SCSI connector and the 50 pin female connector use different numbering schemes. (Only pin #1 on the SCSI connector will be connected to the same pin number on the 50 pin female connector. Pin #2 of the 50 pin female connector will be connected to pin #26 on the SCSI connector, etc.)

Measure 3 inches up the cable from where you installed the first SCSI connector and mount the second one here. Again make sure that pin #1 of this connector is connected with pin #1 of the other two connectors.

13 Measure up 4 inches from this second SCSI connector and mark it with a magic marker. Now count over from the red wire (wire #1) and find the wire #26. Use a razor blade to cut between wire #26 and #27 from this point back to the second SCSI connector. Make sure that you don't cut through to the wire #26 or #27. You are only cutting the insulation between them. You should now have a 3 inch slit in the ribbon cable. Repeat this operation to separate wire #25 and #26. Now cut the wire #26 as far away from the SCSI connector as you can, about 4 inches. Strip 1/4 inch of insulation from it. Cut a 6 inch piece of wire and strip 1/4 inch of insulation from each end. Solder this piece of wire to the separated wire from the 50 wire ribbon cable and cover the joint with a length of heat shrink tubing. Connect the other end of the 6 inch wire to one of the metal tabs on the extra fuse holder (the one that you installed).

14 Identify which of the wires coming from your power supply is the +5 volt line. It will be pin #4 on one of the white plastic connectors that plug into the disk drive and the controller board. Mark it with a piece of tape. Cut a 9 inch piece of wire and strip 1/4 inch of insulation from each end. Solder one end of this wire to the unused metal tab of the fuse holder. Solder the other end of the 9 inch wire to the +5 volt line. The easiest way to do this is to cut the +5 volt line in the middle and strip the insulation from each end. Then solder the two ends of the +5 volt line together with the 9 inch wire from the fuse holder. Be sure to put a piece of heat shrink tubing on one of the wires before you solder them so that you can cover the joint when you are done.

15 Install a 500 milliamp $1\frac{1}{4}$ inch fuse into the fuse holder that you added to the back panel.

16 Connect the two SCSI connectors to the back panel. Use toothed washers between the back panel and the connectors to ensure that the connectors are properly grounded.

17 Plug the 50 pin female connector into the controller board at J4 (see fig. 1). You should check the connection to make sure that pin #1 of the male connector is attached to pin #1 of the female. The pin numbers for the male connector are marked in white lettering on the board.

18 Attach both power connectors, one to the disk drive and one to the controller board (see fig. 1). These connectors are made of white plastic and have four pins arranged in a line.

19 Cut a one foot piece of wire and attach the female AMP connector to one end. Solder the spade connector to the other end. Attach the AMP connector to the ground tab on the disk drive frame. Attach the spade connector to the chassis ground where the power supply is grounded.

20 Screw the rails into the base of the box.

21 Replace the cover to the box.

22 Connect the Mac to one of the SCSI connectors on the back of your hard disk. Connect a terminator to the other one.

23 Cross your fingers.

24 Turn on the power.

Formatting Your Hard Disk

You will need to know the following information about your disk drive before you try to format it. You can find out by calling the technical support folks associated with either the disk drive manufacturer or the place that you bought the parts from. All mail order houses have a technical support phone number for just this type of question. The technical support number for Seagate is (408) 438-5333. I have included the values you will need for an ST225.

Name:	Value used for ST225:	Value input into program:
Number of Heads:	4	4
Number of Cylinders:	615	615
Minimum step pulse:	10 microseconds	12 microseconds
Write current Reduction:	not used	0
Read Precompensation:	300	300
Bad Block List:		

Below you'll find step-by-step instructions on how to use the *U2 SCSI Disk* program to format your new hard disk. Figure 3 includes a list of the possible menu choices. Use it as a reference as you read this section. Before you format your hard disk, you must first enter the information that describes your disk drive. Use the disk drive menu to do this. The Format menu contains commands for the formatting process. They are listed in this menu in the order that they should be executed.

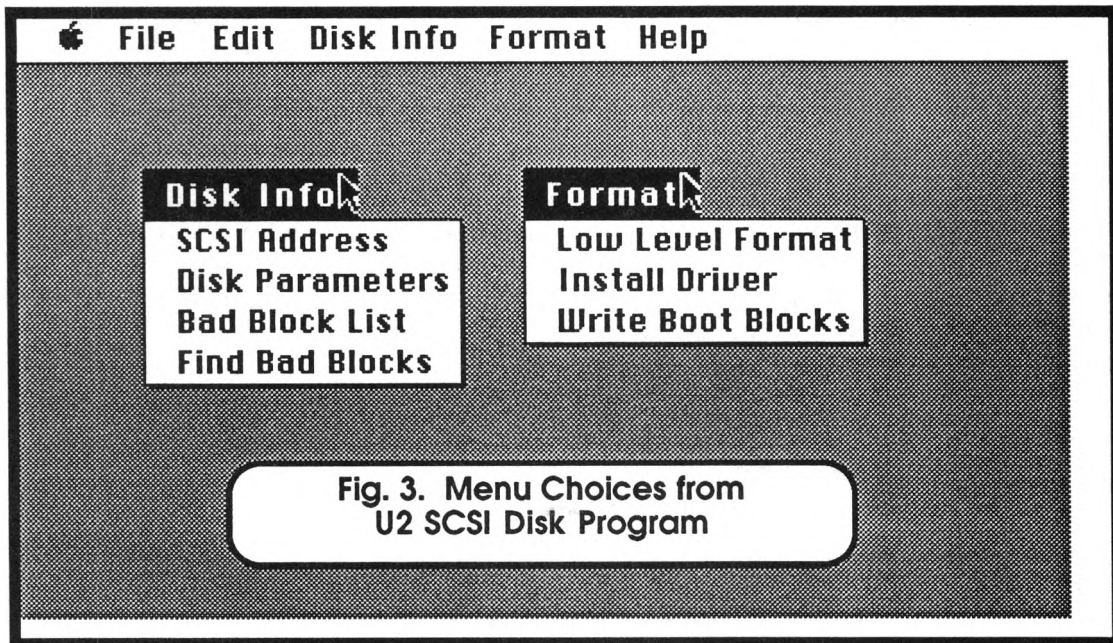
Step-by-step Formatting Instructions

1 Choose the **SCSI Address** command from the **Disk Info** menu. You will need to enter the device number of your SCSI hard disk. This is the number that you set using the DIP switch that you mounted on the back panel. This number is entered in binary form with switch number one being the least significant binary digit. Remember that you can't identify your hard disk as device #7 because this number is reserved for your Mac.

2 Next, choose the **Disk Parameters** command from the **Disk Info** menu. You will need to enter all the information that the controller board needs in order to format your hard disk. This information includes: the Read Precompensation number, the Write Current Reduction number, the step pulse timing, the number of heads and the number of cylinders. You will also have to enter the interleave factor. Use a value of 3 to get the best possible performance.



3 You must now enter the bad block list that came with your disk drive. (If you didn't receive one, skip this step and continue on to step #4. The command entered in step #4 should find most of the bad blocks.) Choose the **Bad Blocks** command in the **Disk Info** menu. Enter the bad block information in the window provided. Notice that you need to enter the cylinder number, head number, and byte-from-index number for each bad block.



4 The next command allows you to find any remaining bad blocks. These will be added to the bad block list which you entered in step #3. Choose the **Find Bad Blocks** command from the **Disk Info** menu. This procedure may take up to 15 minutes as it reads and writes to every byte on the disk at least four times. This step is very important because it ensures that the controller board will not use any part of the disk that contains defects. This is a prime opportunity to go and get yourself another cup of coffee or a beer.

5 Now you are ready to perform low level formatting of your hard disk. Choose the **Low Level Format** command from the **Format** menu. This will permit the controller board to read and write information on the disk drive.

6 The next two steps enable your hard disk to be used by the Mac. First you install a driver program on your disk, then you write the boot blocks. To install the driver, choose the **Install Driver** command from the **Format** menu. Then choose the **Write Boot Blocks** from the **Format** menu to place valid boot blocks on your disk.

7 Now quit *U2 SCSI Disk* program. Use the **Finder** to place a system folder, complete with **System** and **Finder** files, on your new disk. You are now ready to use your SCSI hard disk! Congratulations!

8 Don't forget to buy yourself a back up program. It will be worth its weight in gold when some public domain program accidentally corrupts your hard disk. I think the best back up program is one that is coming out in November called *Trans4™* (from Dantz Software Development, 1510 E-1 Walnut Street, Berkeley, CA 94709, (415) 849-0293).

Making Printed Circuit Boards on a Macintosh

Bryan McDonald

Making a printed circuit board (pc-board) on a Macintosh is easy, inexpensive, and painless compared to the older pencil-and-paper techniques. This article will discuss making pc-boards on a Macintosh using MacPaint, but the technique can be utilized on almost any personal computer. If you have never designed a pc-board on a personal computer, the following guidelines may be helpful to you. (If you have already the finished art work on a personal computer, skip to the section entitled *Making a PC-Board from a Graphic Document*.)

Designing PC-Boards on a Personal Computer

If your pc-board contains integrated circuits (IC's), you will have to draw up a template for a standard IC to use over the entire pc-board. Below is the one that I have found to fit standard 16 pin IC's when using MacPaint. Once you have your template drawn on your personal computer, you should print it out and place an IC over the print out to be sure it fits since many times a graphic object on a screen will look different when printed out.

Some Standard Templates for PC-Boards



16 pin DIP



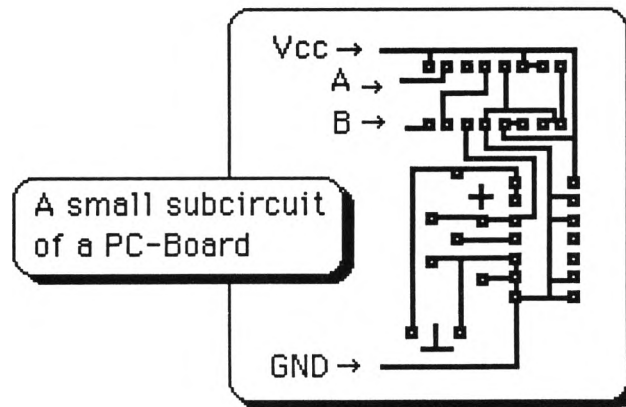
DB-9



DB-25

When creating a pc-board from a schematic, it is good to lay out subgroups of only 2-4 components at a time and interconnect these subgroups later. If the schematic has a power supply containing, for example, a transformer, a rectifier, a voltage regulator, and some discrete components, you should first lay out this sub-circuit and later interconnect the other sub-circuits.

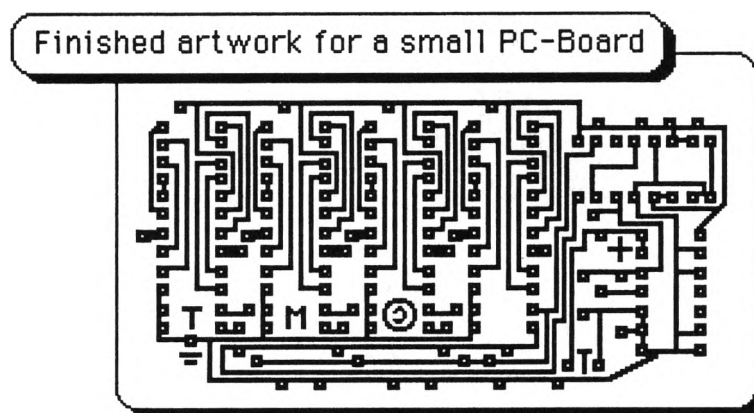
I usually start out my sub-circuits by laying out all the IC's on a blank MacPaint document and connecting everything. If things get too complicated, rearrange the components and start over. When placing IC's it is a convention to have them all facing the same way in rows or columns. A sample subcircuit with input/output lines going off to the left is pictured below.



When starting a layout for a pc-board, leave yourself a lot of extra room for traces (*traces* are the lines connecting everything together) because you will shrink the board down later when all the traces have been routed. I use an Apple LaserWriter so I can sneak by with a two pixel trace width (a *pixel* is short for *picture element*, the Macintosh has 512 by 342 pixels).

I have found that a trace width of one pixel is too thin and gets eaten away in the final process of etching the pc-board. I use a one pixel width between any two traces. This allows me to have five traces fit between an IC and one trace between IC pins (note: these are minimum widths). If you are using a dot-matrix printer, you may have to allow for more room between traces. A guideline is to start out with a two pixel trace width with two pixels between traces.

If you can't seem to fit everything on the board, then you have to resort to jumpers or a double sided pc-board. Double sided pc-boards are easier to draw but are twice as much work to make and solder if you don't have plated-through holes. Hence, a few jumpers on a single sided board are better than a double sided layup (though it is much harder to do a complex circuit without resorting to a double sided pc-board).



If the board seems to be overly complicated, then you might want to start over with the components located differently. Never be afraid to start over! It is best to draw the pc-board as if you were looking from the component side of the board (remember that this will be the inverse

image of the solder side). If you will be writing text on the board, remember to write it backwards since it will be flipped on the final board. If you are working with CMOS, be sure to tie all unused inputs to either Vcc or ground; otherwise the gates may oscillate at the chip's frequency and use much more power than a stable chip.

Summary

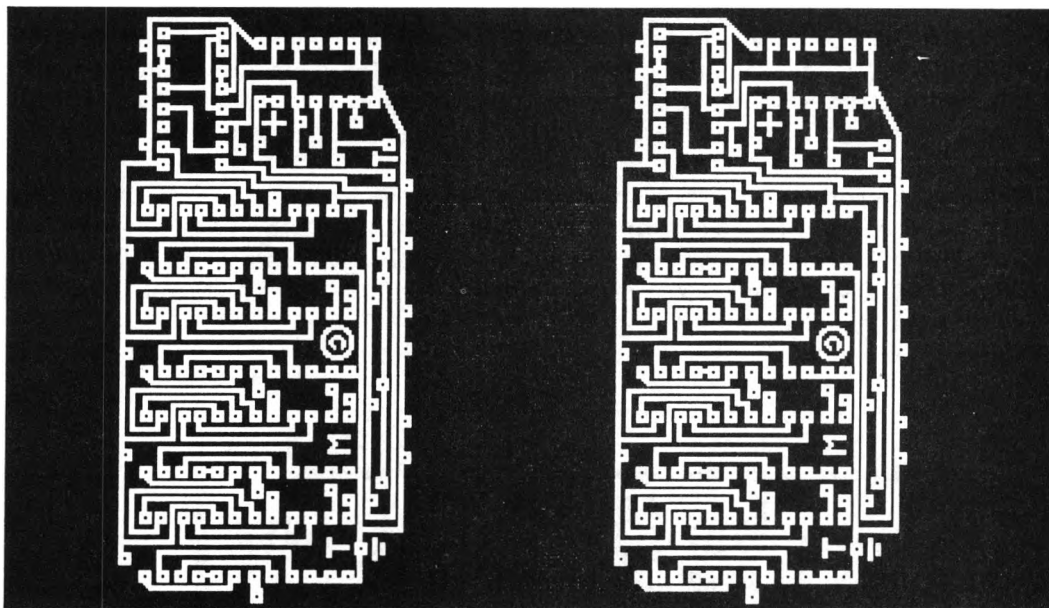
In summary, here are the rules I follow when making pc-boards:

- Rule #1: Never be afraid to start over.
- Rule #2: Don't use traces less than two pixel wide.
- Rule #3: No more than one trace between IC-pins.

If you want to make a pc-board from a picture of the finished art work, digitize a picture of it using say, Thunderscan, and touch it up following the above rules.

Making a PC-Board from a Graphic Document

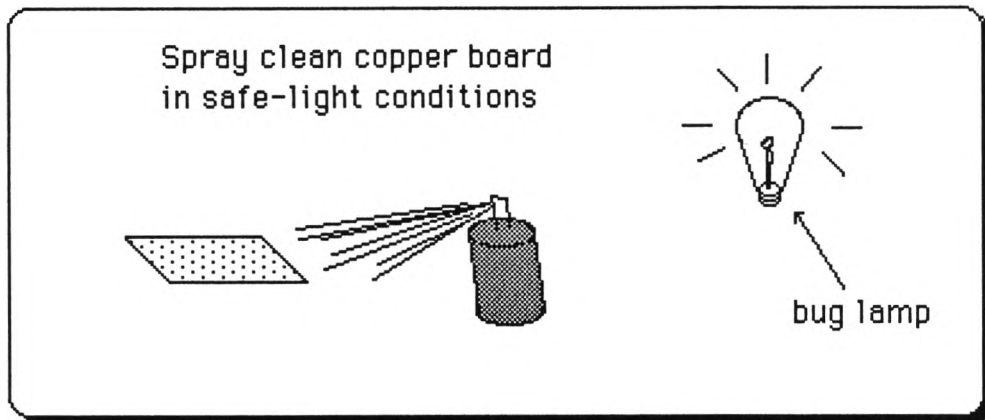
When fabricating a pc-board using photographic techniques, you need a negative of the artwork to expose the photosensitized copper board. Using your graphics art program, invert your art work and make two printed copies of it:



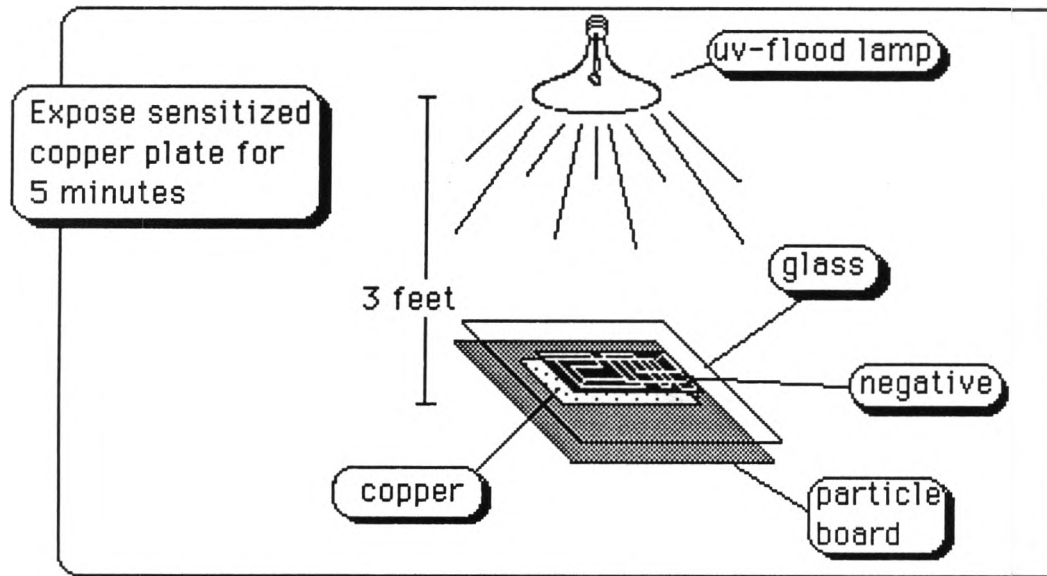
Hold your printed output up to the light. You will notice that the black is not very dark and could be a lot darker. You can easily solve this by making a five-cent copy of this printout at the nearest copy store. The copy will be much darker. Then make a transparency from the black copy (almost any copy store can make a transparency for less than a dollar), and you will have an inexpensive negative of your pc-board. Hold the transparency up to the light and notice again how the black is not very dark. To solve this, cut the transparency in half and lay the two halves on top of each other, lining up the pc-boards. Holding it up to the light, you will notice a big improvement in the darkness. This two-layer transparency should have sufficient darkness. This method for making negatives is about \$14 cheaper than that of a photo shop!



You are now ready to sensitize the raw copper board. Cut out the copper board and clean it with Ajax until it is spotless and shiny. Dry off the board and do not touch the copper anymore. Go into a room with no ultra-violet light present (a dark bathroom with a bug light is ideal) and spray on a layer of Photo-Resist Liquid (Dick Smith Cat ER-71) onto the copper. Let each layer dry over night or in a warm oven for 20 minutes. If you don't let the resist dry long enough, it won't hold up in the etching process. I usually need about two coats of the sensitizer to get the copper looking even and shiny.



When the board is dry, you are ready to expose it. Place the transparency over the sensitized board and put it between a sandwich of glass and particle board so it won't move while being exposed (instead of the particle board any hard flat surface could be used). Place it under an ultra-violet floodlight (Sylvania makes one called the R2 SUPERFLOOD DXC, available from Dick Smith Electronics Cat S-3905) for five minutes at 3 feet away. You may use sun light as your ultra-violet light source, but you will have to experiment to find the ideal time of exposure. I have heard that 2 minutes is enough time for a cloudless fall day in Berkeley. If you under expose the board, some of the traces may be eaten away in the final etching process. If you over expose it, some light may seep in around the traces.



Now it is time to develop the board. Carefully take the board out from the glass-particle board sandwich and put it in a bath of Photo Etch™ developer (Dick Smith Cat ER-8) for 35-45 seconds. Be sure to agitate the bath every 5 seconds for an even exposure. Take the board out and **let it dry face up!** The directions on the back of the can of the developer say to put the board face down in tissue, but I find that this scratches the board. When the board is dry, place it in the developer bath again for 10 seconds and don't agitate it (the directions on the can say to use a fresh bath of developer, but I have found that the original bath can be used). When the board is completely dry, you may bring it into normal light again.

Once the board is dry, you are ready to etch it in a copper eating solution. I use a Pyrex pie plate filled with Ferric Chloride Etchant (Dick Smith Cat H-5652 or Radio Shack Cat 276-1535) and heat it to near boiling before putting in the board. With warm solution it will take about 7 minutes to etch the board I have illustrated (note: this time may vary). The more you agitate the solution, the quicker the copper will get eaten. You may take the board out periodically to see if all the unwanted copper is gone. Once you think that the board is in its final form, you may put it under running water to wash off the etchant for a closer look. If the board needs to be etched more, put it back in the etchant.

BMUG
Fall 1986



Place the developed pc-board in heated etching solution to shorten the etching time.



You should end up with a professional-looking board costing from \$5 to \$10 per board. Don't get discouraged if your first board does not turn out— it is a fairly complex process to get perfect the first time. Remember that you would have paid up to \$300 to have a pc-board company make the board for you.

Bryan McDonald is a student at U.C. Berkeley studying Applied Mathematics and is also a technician at Meeting Technologies, Inc. He welcomes correspondence and can be contacted at 2705 Benvenue Ave. Berkeley, CA 94705 (415)-548-8422, on CompuServe 72106,3232 or on the BMUG BBS.

Msg. #2004 in **Public Messages** Posted on 07/06/86 at 13:42:39
To: ALL From: DEAN BENZ
Subject: Make a Mac+ hard drive question
AND NOW.....

The Challenge of the week....

I understand that it is fairly simple to hook up a Hard Disk to the MAC+ SCSI Port. The Way I understand it is...

1. Get an off the shelf hard drive at your local electronics surplus.
2. Hook a SCSI Disk controller to the drive.
3. Hook the controller to the MAC+ SCSI port (custom cable?)
4. Tell the Mac that there is a H.D. (modify H.D. 20 DOC.?)
5. format dialog should magically appear, and your on your way.

Sounds easy enough, I can make the cable etc. but how do I tell the system that the Drive exists? Has anybody out there made their own Hard Drive? I don't expect that the 3rd party drive manufacturers are going to tell me so I'm hoping there is someone at Apple, or in the population at large who can help.

Thanks in advance.

SCSI Hacking

Michael Lamoureux

The Small Computer System Interface (SCSI) port on the Macintosh Plus provides the Mac users access to a large number of fast, intelligent peripherals like hard disks, tape backup, file servers, and so on. However, it also gives the serious hardware hacker access to a fast data bus into the Mac, giving the chance to hang a mess of wire out the back of the computer attached to any hardware device imaginable. This article describes how to control the SCSI port for use as a general purpose I/O port.

The SCSI connector on the Mac Plus is a 25 pin port, of which 18 can be used to either send or receive data and control the flow of information (See **Figure 1**). Although a true SCSI device attaches special meaning to each pin in order to coordinate the transfer of data, for our purpose it is best just to control each pin as we please and ignore any special interpretation of the SCSI standards. For instance, the pin marked BSY is used by a true SCSI device to see if the port is BUSY, but we can set this pin anyway we like.

Tutorial

The SCSI port on the Mac Plus can be controlled by reading and writing to specific locations in memory. Below is a quick and dirty description of how to control each pin on the SCSI connector shown in figure 1, written in BASIC syntax with PEEKs and POKEs. Connecting a voltmeter to the appropriate pins on the back of your Mac Plus allows you to monitor each output. Obviously, don't try this with a hard disk or anything other true SCSI device attached to the SCSI port!

To initialize the SCSI port, execute:

```
POKE 5767169 + 16*2, 64      (Puts chip in target mode)
```

To read a value present on the SCSI data bus (pins DB0-DB7), execute:

```
POKE 5767169 + 16*1, 0      (Puts data bus in input mode)
PEEK(5767168)                (Reads in the data)
```

To write a value onto the data bus (pins DB0-DB7 & parity DBP), execute:

```
POKE 5767169 + 16*1, 1      (Puts data bus in output mode)
POKE 5767169, V              (writes the value V, with
                              V between 0 and 255)
```

To control pins REQ,MSG,C/D and I/O on the SCSI port, execute:

```
POKE 5767169 + 16*3, V      (V between 0 and 15)
```

To read the inputs ATN and ACK, execute:

```
PEEK(5767168 + 16*1)
```

and examine bits 1 and 4 of the value returned.

To control pins SEL and BSY on the SCSI port, execute

```
POKE 5767169 + 16*1, V      (V between 0 and 15)
```

For instance, V=8 sets the BSY bit, V=4 sets the SEL bit and V=12 sets them both. This also puts the data bus DB0-DB7 in input mode; odd values of V puts the data bus in output mode.



Description

This tutorial really covers all you need to know, though it might seem very strange at this point. What the commands above are actually doing is reading and writing to a set of registers inside a special chip in the Mac which controls the SCSI port. This chip, called a NCR 5380 SCSI Interface device, has seven registers which are used to send and receive signals over the port. The Mac treats the registers on the SCSI chip as locations in memory and by using PEEKs and POKEs you can read or write to any register.

To read register #A on the SCSI chip, you would execute:

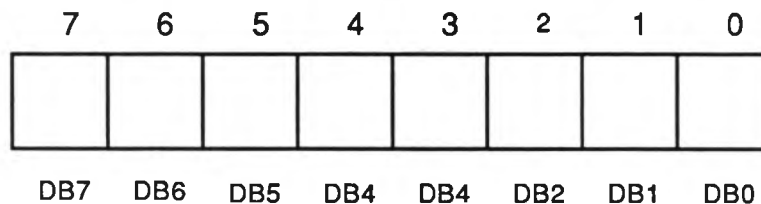
`PEEK(5767168 + 16*A)` where A = 0, 1, 2, 3, 4, 5, 6, or 7

To write a value V into register A, you execute

`POKE 5767169 + 16*A, V` where A = 0, 1, 2, 3, 4, 5, 6, or 7
and V is between 0 and 255

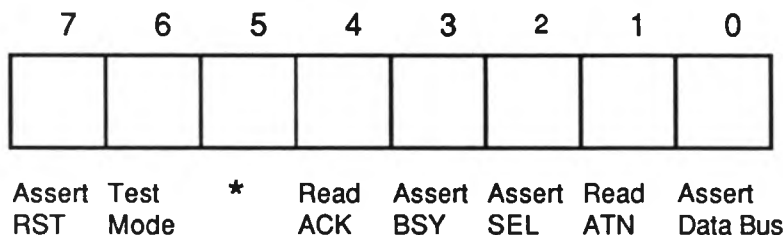
We need only use four of the registers and for simplicity, I will assume the chip is used in "Target mode" which just means that register 2 is set as below. Here is a description of each register that is used in the tutorial and what each bit in the register does; examine the commands in the tutorial to see how each register was used.

Register 0:



Controls the data bus of the SCSI port, pins DB0 to DB7, along with the parity bit DBP. Reading this register reads the data on the corresponding pins of the SCSI bus while writing to the register sending data out to the same pins, with DBP reflecting the parity. This output data must be enabled by setting bit 0 in register 1 (Assert Data Bus), and the output data is inverted.

Register 1:



This register is harder to explain.

Bit 7 (Assert RST) resets the chip, putting zeros in all registers and outputs a low on the SCSI port, pin 4 (RST). Usually you just leave this bit at zero, since the Mac automatically resets the chip for you upon rebooting.

Bit 6 (Test Mode) disconnects all outputs of the chip. Although that may be useful for some reasons, usually you just leave this bit at zero.

Bit 4 (Read ACK) and **Bit 1** (Read ATN) reflect the values input on the corresponding pins of the SCSI bus. Writing to these bits has no effect.

Bit 3 (Assert BSY) and **Bit 2** (Assert SEL) control the corresponding outputs on the SCSI bus. Writing a one puts a low on that pin, while a zero puts a high out. Reading these bits just tells what you are currently outputting.

Bit 0 (Assert Data Bus) controls the direction of data flow on the SCSI pins DB0-DB7 and DBP. Writing a one here sets up those pins as an output, while a zero turns them into an input. This is important; if you try to input data while this bit is not reset correctly, all you read is garbage.

Register 2:

7	6	5	4	3	2	1	0
0	1	0	0	0	0	0	0
*	Target Mode	*	*	*	*	*	*

This register puts the SCSI chip into target mode, which you normally do at the initialization step. Write a "64" (decimal) in here at the start of your program and then leave it alone.

Register 3:

7	6	5	4	3	2	1	0
*	*	*	*	Assert REQ	Assert MSG	Assert C/D	Assert I/O

This register controls the output pins REQ,MSG,C/D and I/O. Writing a one in any of the bits puts a low on the corresponding pin of the SCSI bus. A zero puts a high.

Registers 4, 5, 6, 7:

Just ignore them!

Additional Technical Information

All outputs are inverted while inputs are not. That is, if you set a bit to one in a register that controls and output pin on the SCSI port, the pin goes low. However, if you place a low signal on an input pin, the corresponding bit in the register is zero.

All the outputs on the SCSI port, except RST, are "open collector" which means they will sink current but not source it. This is fine if you are connecting TTL devices to the port but otherwise you may require pull-up resistors. See the application below for an example.

The RST (Reset) pin on the SCSI port is a little different than all the others. It already has a 10K



pull-up resistor attached inside the Mac so normally you can measure +5 volts there, which can be used by the attached hardware device to sense when the Mac Plus is on. (Don't use it as a power source!) Also note that you can set this pin to 0 volts by writing a one in bit 7 of register 1— but this resets the whole chip! Finally, the Mac pulses this line low twice on reboot, as it tries to locate any SCSI devices.

The SCSI port can also be used in "Initiator mode" in which case the pins BSY, SEL, REQ, MSG, C/D and I/O become inputs, while ATN and ACK become outputs. For a complete description, get the data sheets for the NCR 5380 SCSI Interface Device.

Application

The point of all this information is that you can now get signals in and out of you Mac very quickly with the SCSI port and so you can use it to control any number of hardware devices even down to the chip level. The port is very fast, allowing you to send out pulses as short as one microsecond and transfer bursts of data at a top speed of 1,000,000 bytes per second. As an example, look at **Figure 2**. Here, I have a high speed 8 bit analog to digital converter chip (National ADC0820) connected directly to the SCSI port that runs at 200,000 samples per second. The SEL output of the SCSI port is pulsed to start the data conversion which is completed in under a microsecond, at which time the Mac reads in the data through the input bus DB0 to DB7. Notice the 1K pull-up resistor on the SEL line which is needed since the 0820 CMOS chip expects to see a solid +5 volts on its input, which the SCSI port does not provide with its open collector outputs. I used a value of 1K for a fast rise time; 10K would be acceptable in other cases.

The essentials of the routine to read in data are as follows:

```
#asm
    move.l    #(count),D0          ; count= # of bytes to input
                                      ; keep count < 65000
    movea.l   #(ptr),A3           ; ptr = where to store the data
                                      ; i.e. points to a big buffer
    movea.l   #(5767168),A0       ; SCSI reg 0 address, read
                                      ; needed to read in data on SCSI bus
    movea.l   #(5767169+16*1),A1  ; SCSI reg 1 address, write
                                      ; needed to control SEL line
    movea.l   #(5767169+16*2),A2  ; SCSI reg 2 address, write
                                      ; needed to initialize SCSI port
    move.b    #0,D1              ; value to set SEL line high
    move.b    #4,D2              ; value to set SEL line low

    move.b    #64,(A2)           ; initialize SCSI port to target mode

loop:
    move.b    D2,(A1)            ; set SEL low (start of pulse)
    move.b    D1,(A1)            ; set SEL high (end of the pulse)
    move.b    (A0),(A3)+         ; fetch and save the data
    dbf       D0,loop            ; repeat til done
#endasm
```

This loop takes five microseconds to execute which is the limiting factor on the speed of this A/D conversion setup. A better design of the hardware connections could increase the speed, but I like this one-chip arrangement because of its simplicity.

Other Applications

I was motivated to write this article after playing with a device made by Chad Pennebaker of Douglas Electronics (San Leandro, CA). He has designed a couple of general purpose I/O ports that attach to the SCSI port which are controlled with PEEKs and POKEs along the ideas above. He uses them to control a pen plotter and other equipment at Douglas, increasing the automation of their production operation, yet each device only costs around \$40. The board I used has 24 pins for input and 24 pins latched output, providing plenty of control for external devices. I also had a couple of fast analog to digital converter chips lying around at home which I had never been able to use because I had no way to control them fast enough, until I used the SCSI port. Obviously this could be done with any microprocessor-compatible hardware chip, so you may as well connect multi-channel A to D, D to A chips, hardware multipliers, EPROMs and so on to your Mac. Although it is not yet an "open Mac", the Macintosh Plus with its SCSI port may be close enough.

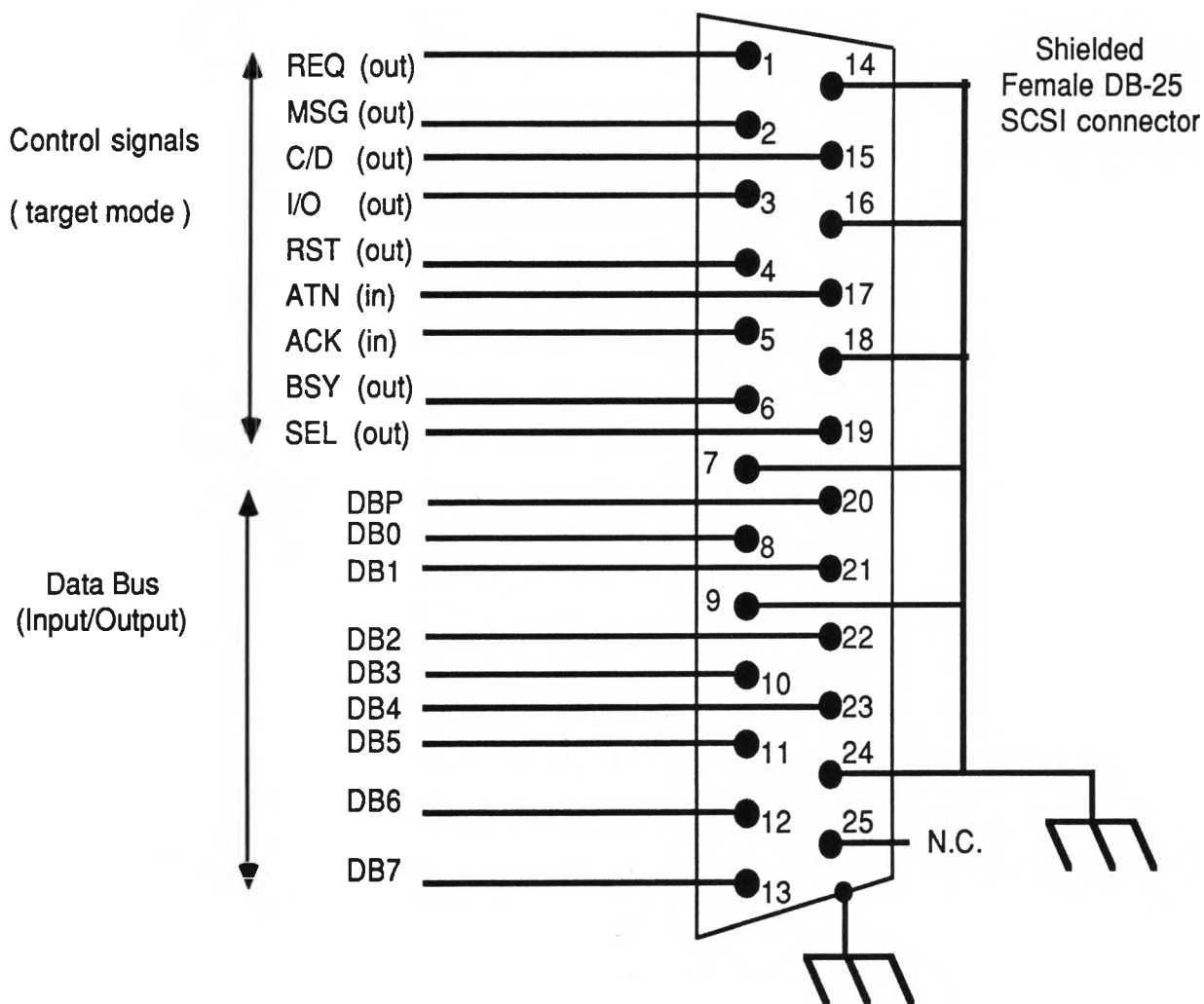


Figure 1

Typical voltage source
(analog signal being digitized)

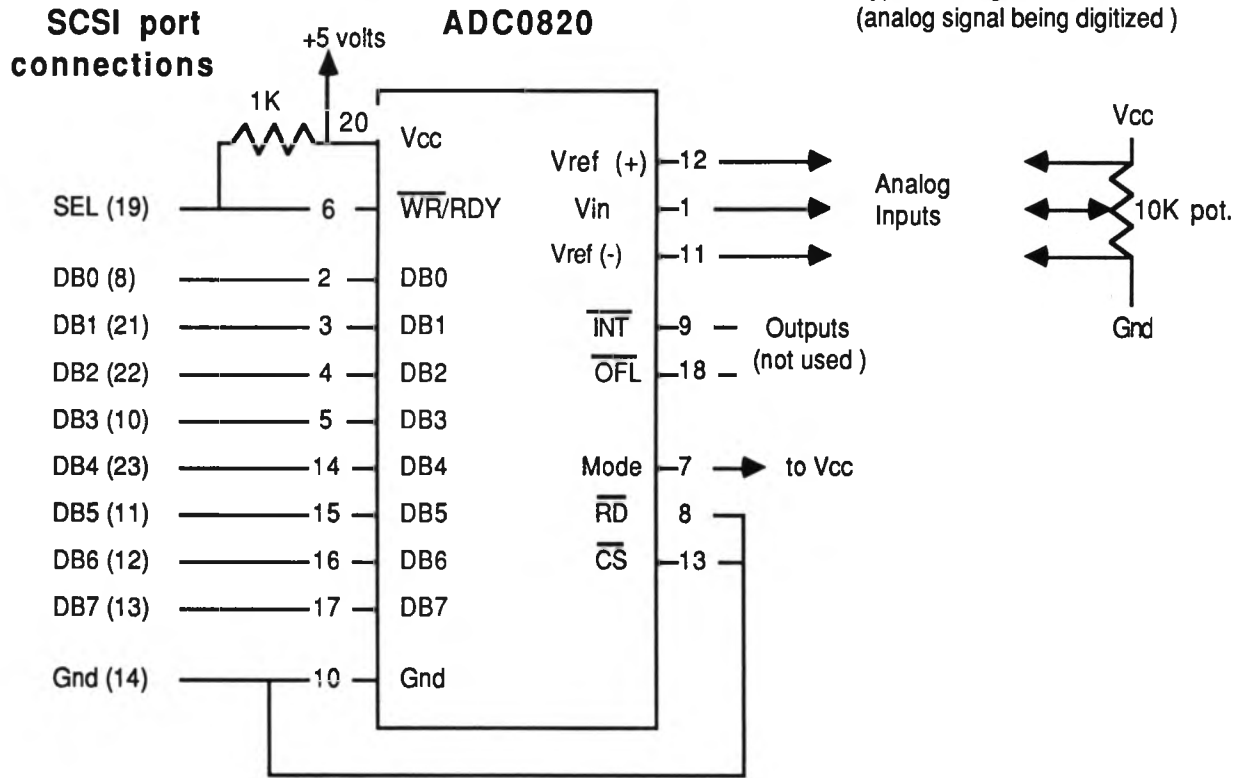
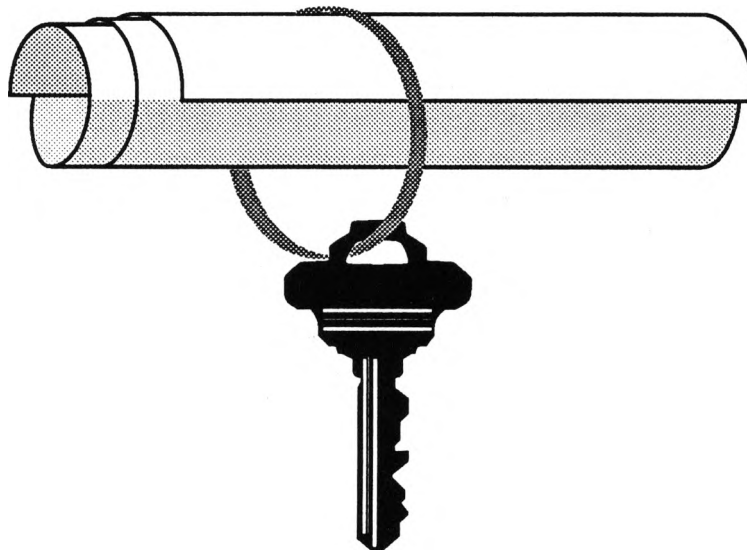


Figure 2



Scroll/Dan Wood

Adjusting Your Mac

David W. Berry

Lately I have been seeing a growing number of articles and requests concerning how big the Mac screen image should be, how to adjust the Mac's screen and similar questions. This article outlines how those adjustments should be made. The assumption is made that whoever performs this feat has at least enough electronics experience to be able to read a volt-meter and make simple adjustments

The following tools and/or equipment are needed to perform the procedure given here:

Torx Screwdriver	Available a wide number of places. Needed to open the Mac case.
Metal Ruler	
MacOpener	Or some other way of prying the Mac case apart once the screws have been removed
Small Screwdriver	Bit should be about 3/32" across. Metal will work but plastic would be better.
TV Adjustment Tool	Small plastic hexagonal tool. Available from Radio Shack. This one <u>Must</u> be plastic.
Voltmeter	Preferably digital. Will be used to adjust the +5 Volt power supply.
Common Sense	Enough to know not to grab things like the video tube. We're working around very high voltages here.

Making the Adjustments

Opening Your Mac

- A. Turn the Mac off.
- B. Lay your Mac face down on a towel.
- C. Remove the 5 Torx screws. One is hidden behind the battery cover. Put the screws and the battery cover somewhere you won't lose them.
- D. Use the "Mac Opener" to open the Mac case.
If you don't have a "Mac Opener" you can use a stiff metal ruler to pry the two halves of the shell apart. Be very careful as it is quite easy to damage the appearance of the case.

Adjust the +5 Volt Power Supply

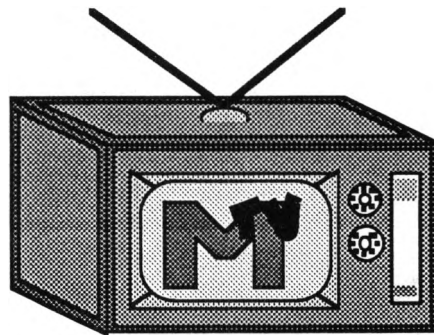
- A. Set the Mac upright on a non-conductive surface in a position where you can easily reach the back and left sides without stretching over the Mac.
- B. Find the power connector on the logic board.
- C. Attach the ground of your meter to the 5th pin from the rear.
- D. Attach the positive lead of your meter to the pin at the rear.
- E. Set your meter to measure +5 VDC.
- F. Turn the Mac on. The meter should read in the neighborhood of +5 VDC. If it doesn't, turn it off and go back check the connections.
- G. Adjust the "Voltage" pot on the left side of the Mac.
 1. Turn it up until either:
 - a. The meter reads 5.2 VDC, or
 - b. The power supply "crowbars." This is a protective feature indicated by the screen going black and a chirping noise.
 2. If you were able to adjust the supply to 5.2 VDC without it crowbarring, adjust it to exactly 5.0 VDC.



3. If the power supply crowbarred before you reached 5.2 VDC then turn the Mac off and back on to clear the crowbar condition and adjust the power supply to .1 VDC less than the point at which it crowbarred.
4. If the power supply cannot be adjust above 4.8 VDC without it crowbarring you have a defective part in the power supply and should take it to a dealer to get it repaired (and presumably adjusted too.)

Adjust the display. There are four adjustments for the display which should be made until repeatedly until all are satisfactory.

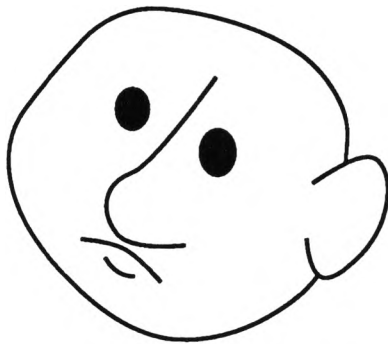
- A. Width - do I really have to say? Make the screen 7.11" wide with this one.
- B. Height - uh, this one controls screen height. The screen should be 4.75" high.
- C. Focus - Since the face of the Mac screen isn't all the same distance from the electron gun, you must choose a part of it to be in focus. The best thing to do is to get a point halfway between the middle and the outside focused as sharply as possible.
- D. Cut-Off - This is almost a contrast control. It should be adjusted so that the intensity control on the front panel has a reasonable range. This is how I do it:
 1. Turn the screen intensity all the way down.
 2. Turn the room lights way down.
 3. Adjust the Cut-off until you can just barely see the Mac screen.



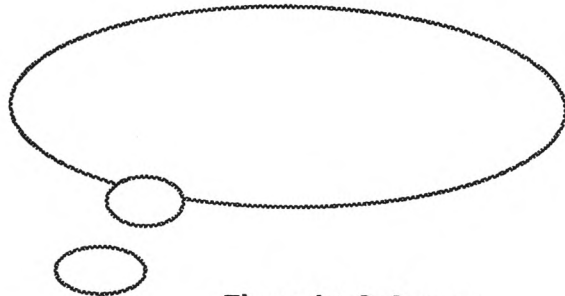
TV /Wood



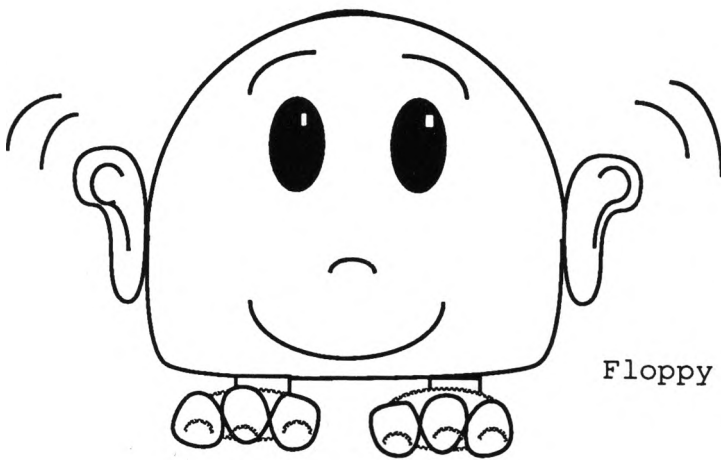
BMUG
FALL 1986



Brooklyn
Baby



Thoughtfulness



Floppy



Roger

Copyright © 1985 Dan Wood

BMUG
Fall 1986



Connectivity

CONNECTIVITY

AppleTalk Networking • Large-scale corporate networking	Michael Rogers	303
MACLINK-ing CP/M and MS DOS • Wordstar: From IBM to Mac and back	Kenneth William Hirsch	310
MacLink: IBM Data Transfer to Mac • Simple data transfer from an IBM PC to a Mac	Harry Critchfield	313
The Mac Underground • Connecting to an on-line intelligence system	Chris Halaska	314
CompuServe Update • Dungeons and Dragons On-Line	Linda Custer	320
2400 Baud Modems • Experiences with faster modems	Timothy Standing	322
Strategic Conquest Plus • Connecting with a Mac game	Steven Fransen	323
MCS: Multichannel Communications System • Transatlantic transmission of files using Macs	Paul B. Blood	326
Two for the Lab: MacADIOS and LabVIEW • The laboratory interface grows up	Paul F. Daley	328



AppleTalk Networking

Michael Rogers

Since I'm involved in a project to install a medium-to-large scale network of Macintosh computers, I thought I'd share some of my experiences. The network I'm working with has, at last count, 32 Macintoshes, 14 LaserWriter Pluses, 4 ImageWriterIIs, and 11 IBM PCs growing all the time. The attached diagram shows how these various devices are distributed among 13 distinct AppleTalk networks which are then bridged together to allow communications between any two devices for file transfer, electronic mail, remote printing, etc.

The networks were set up based on geographical location - Macintoshes and IBM PCs near each other (same building or same floor of a building) are networked together to share a LaserWriter. There is at least one LaserWriter Plus in each location. To allow communications between nodes on different networks, some of which are separated by as much as nearly two miles, these individual networks are then bridged together using a network in a central location which does nothing but bridge AppleTalk networks.

The computers are typically located not on someone's desk, but in centralized computer rooms, available to anyone in that particular department. The computers are shared among almost 200 people, and the 32 Macs currently available will likely need to be greatly expanded. The hardware in each location has been standardized as much as possible. The Macintosh workstations each have a Macintosh Plus, a 20-meg hard disk (either Apple's HD20 or HD20SC or a DataFrame 20). The hard disks are used only for system files and applications; all documents are now kept on floppy. The software at each Mac varies depending on the needs of the location, but as a minimum includes Microsoft Excel, Microsoft Word, and MacDraw. The IBM PC workstations are not so standardized, but typically are a PC AT with 512K or more of memory. All LaserWriters are Plus models and all ImageWriters have the AppleTalk option installed.

The following sections of this article cover typical networking considerations and the pros and cons of the various options available.

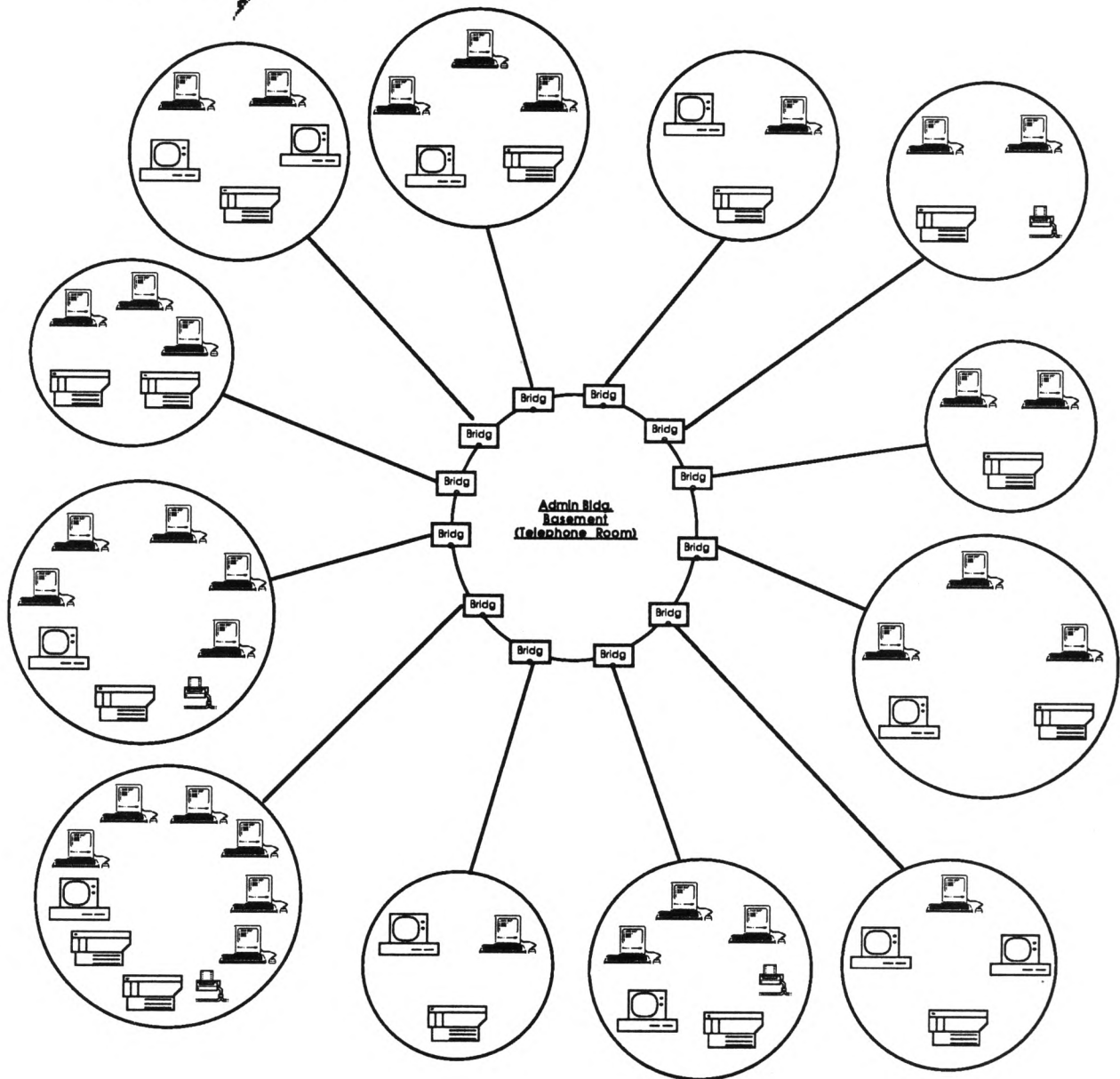
Networking Hardware

At this time, there are only two viable options for AppleTalk hardware: Apple's own connectors and cabling, and the PhoneNet system from Farallon Computing in Berkeley. Apple's hardware consists of an AppleTalk connector, a box that allows network data to travel both directions from the node and has a cable connecting to the node (either a circular 8-pin connection for a Mac Plus or ImageWriter 2, or a DB-9 connection for a 512K Mac, LaserWriter, IBM PC, or Hayes Interbridge). The network cabling itself is a non-standard 3-wire system in a round, shielded cable. AppleTalk is limited to a total distance of 1000 feet of cable and 32 nodes.

The PhoneNet hardware offers some considerable advantages over Apple's hardware. PhoneNet is 100% AppleTalk compatible from a software standpoint and is superior from a hardware standpoint. PhoneNet has a box similar to Apple's described above, but the cabling uses standard modular phone cable. Because of the hardware differences, PhoneNet appears to be more stable when adding, removing, or powering up or down network nodes. PhoneNet has also been tested at distances up to 8000 feet successfully. Farallon is also promising a network repeater which will allow network runs of almost any length. Because PhoneNet uses standard cabling, networks can be installed less expensively than with Apple's hardware.



BMUG FALL 1986



APPLETALK NETWORKS

	Mac Plus w/ 20-meg hard disk, Microsoft Excel, Microsoft Word, MacDraw, TOPS, Spellswell, HFS Locator Plus, other applications, modems as required.
	IBM PC w/ TOPS for the PC
	Laserwriter Plus
	Appletalk Imagerwriter 2 (Color Dot Matrix)
	Hayes Interbridge, to bridge Appletalk networks

10/17/86 MWR



PhoneNet offers cost savings in another very important way. If you are going to network computers in different rooms, floors, or even different buildings and you have a modern phone system, PhoneNet may already be installed! Modular phone cables have four wires, only two of which are used for the phone signal. PhoneNet uses the spare two wires and allows you to use the phone and the network simultaneously. By going to your central phone room and jumpering the PhoneNet pair of wires from the telephone cables together for the rooms to be networked you have completed the installation. Just plug a modular phone cable from the PhoneNet connector into the wall and you can communicate. Add a splitter if you need to also use your phone or a modem. Two notes of caution: a resistor (provided with the PhoneNet connector) may need to be placed across the network wires at the end of a long network run to reduce reflections (these resistors are built in to the Apple connectors but not into PhoneNet), and if you plan to use a modem on the same line, disable the AppleTalk wires out of the modem as some modems will put information on these lines (for use only with special phones).

I have been quite happy with the PhoneNet system and recommend it over Apple's hardware. If you have Apple hardware in place, Farallon also sells an Appletalk-to-PhoneNet adaptor cable.

Network Traffic/Bridging Networks

Network traffic can build to the point of bogging down the network if you are not careful. Although AppleTalk has a software limit of 32 devices per network, you probably don't want anywhere near this number on any one network. Our largest network has 10 nodes. We chose to create small networks, defined by geographic location, to keep the traffic on any one network to a minimum. Network traffic will build especially fast if you have multiple printers on the same network, have an E-mail system running, or are using a file server or multi-user applications. AppleTalk is a medium-speed network (running at 230Kbaud) and can become quite busy.

If you find that your network is bogging down, or you have more than 32 nodes on your network, but still want remain flexible for communications between any two devices, consider splitting the network into two or more networks and bridging them together. One device available for doing just this is the Hayes Interbridge. The Interbridge splits one network into two and isolates them from each other in most circumstances. If you have two LaserWriters, for example, you may want to put one on either side of the Interbridge. By doing so, printing to one of the LaserWriters does not affect the other network. When you need to access a printer or other device on the other side of the Interbridge, you can still do this; Interbridge passes information from one network to another when necessary.

Looking again at the network drawing, the 12 circles around the center circle represent 12 different AppleTalk networks, each with different combinations of printers and computers. The center network has nothing but Interbridges on it and serves to connect all of the 12 other networks. This network of bridges is located in the basement of our main building. The entire phone system is routed through the basement, so we can establish communications with any location throughout the company from this one location. With an Interbridge for each of the networks, a device on any network can communicate with any other device in the system. This all happens invisibly to the user. When someone calls up the Chooser desk accessory and selects the LaserWriter icon, a list of 14 LaserWriters, all named according to their location, is presented. Any one of these can be printed to, over distances of 2 miles in some cases. To simplify matters when a large number of networks are defined, the system can be divided into "zones". A zone would be a subset of the entire system, so that the list of available LaserWriters, for example, would be smaller and might represent the printers that the people in one



BMUG FALL 1986

department might reasonably be expected to want to print to. Another desk accessory called the Zone Chooser then allows you to switch zones for the rare occasion when you need to communicate outside your zone.

For long-distance communications or for communications where there is no direct AppleTalk or PhoneNet connection, the Interbridge also allows you to communicate over phone lines and modems to remote networks. This requires an Interbridge and a modem on each end of the connection, and communications take place at modem speed (up to 19.2Kbaud), not AppleTalk speed. It is possible using this configuration, for instance, to choose the print command at your Mac in Cupertino and have your output appear off a LaserWriter in Armonk, New York. Each Interbridge can handle one local bridge and two remote bridges.

Bridges from AppleTalk networks to other industry-standard networks are beginning to appear. Kinetics in Walnut Creek offers the Fast Path, a bridge from AppleTalk to Ethernet which appears to work quite well. This option offers the possibility of bridging AppleTalk networks by using an Ethernet backbone already in place in many corporations. Another recently announced use of such a bridge are multiuser applications residing on DEC VAX machines. Odesta Corp. has announced a multiuser version of its Helix database which resides on a VAX and communicates with Mac workstations using the Kinetics FastPath Bridge.

A recent announcement by Apple Computer and Northern Telecom will allow Macintoshes to be networked to IBM PCs, IBM System 36/38 and mainframe systems, DEC, HP, and Data General computers. Network data will be routed through Northern Telecom's Meridian SL-1 phone system and will travel at 19.2Kbaud. The SL-1 just happens to be the central phone system mentioned above. The system will also support communications with outside services and AppleLink for electronic mail and messaging through a shared pool of modems.

For network access to IBM mainframe computers, Tri-Data has a product called the Netway 1000A. The Netway 1000A is a network device rather being attached directly to a single computer. As such, anyone on the network can use it for mainframe access. In fact, the Netway 1000A supports 16 simultaneous connections.

Networking Software/File Service

To fully utilize the network once it is in place, you'd probably like to use it for more than just sharing printers. Centram Systems of Berkeley has a network software package called TOPS which allows every storage device on the network (floppy or hard disk) to become a file server for the network. With the TOPS software loaded on each machine in the network, any drive can be "published" to the network making it available as a file server. Users on other machines in the network (or on other networks if you have a network bridge) can access files on that drive. This means that they could edit a file stored on another machine using an application on their machine, they could run an application stored on the remote drive, or they could transfer the file or application to their machine for use locally. One can also create files on the remote drive by creating a new document on the drive or by transferring a file or application to the remote drive. By "mounting" a remote drive on your local Mac it appears on your desktop and in the standard open and save dialog boxes as if it were physically attached to your Macintosh.

TOPS also offers a number of protective options, including prohibiting simultaneous use of a file by more than one user, password protection of drives, and read-only protection. If you feel you need a centralized file server for document storage, a large capacity hard drive attached to one of the Macintoshes with a network running under TOPS may work quite well.



If you don't like IBM PCs, skip this paragraph. TOPS also allows an IBM PC to join the AppleTalk network. This offers a number of advantages if you are trying to bring Macintoshes into an IBM PC dominated environment, as I was. Whereas TOPS for the Mac involves only software, TOPS for the IBM PC consists of software and a hardware card (of course!). The card slides into one of the slots in the PC and allows a standard AppleTalk or PhoneNet connection to be made. Once the software is loaded onto the machine, the IBM just appears to be another storage device on the network from your Mac. If an IBM drive is published, Mac files and applications can be stored there (and vice versa, IBM files can be stored on Mac floppy or hard drives). This offers a convenient way to bring IBM format documents into the Mac world. For example, publish your hard drive on the IBM PC, call up Excel on the Mac, and open each of the Lotus 1-2-3 files on your IBM hard drive. Excel converts them automatically to Excel files and you never have to use Lotus 1-2-3 again! This sort of capability should be very important when Ashton-Tate's dBASE Mac is released, since it will be able to read existing dBase III files. TOPS also allows IBMs to communicate with LaserWriters. If an IBM PC package supports either PostScript or the Diablo 630 printer, TOPS will put the LaserWriter in the proper mode and print to it. An unreleased product now in beta test will allow any IBM program which can print to an Epson FX-80 (almost all can) to have its output translated into PostScript and sent to the LaserWriter.

TOPS is not the only networking software available. Another popular product which offers similar features is MacServe from InfoSphere. MacServe does not offer quite the flexibility of TOPS and does not include options for IBM PCs, however. Since I have not tested MacServe extensively I will not say anything more at this point.

Dedicated file server hardware is a wide-open market for the Mac. In other words, not much is available. The industry still awaits Apple's long-promised AppleTalk file server. The rumor mill recently has had lots of info about Apple's file server, so they may be getting close to actually releasing it. At present, however, the only option for dedicated file server hardware is from 3COM. 3COM's 3Server combined with software called Ether Mac make up the system. The 3Server can range from 36 to 420 megabytes of storage partitioned into volumes with varying access rights and password protection. The 3Server also claims to offer LaserWriter print spooling through a serial connection on the 3Server itself but I have not seen this demonstrated. The 3 Server appears to fit in well in an office which already has IBM PCs linked with 3COM's EtherLink board over Ethernet. Adding Macintoshes to the network so that both Macs and PCs can share the 3 Server is possible. The administrative functions required of a network administrator can only be performed from an IBM PC, so it appears that using the 3Server in a Mac-only environment is not a preferable option. My recommendation for dedicated file server hardware is to wait.

Print Spooling

Printing to the LaserWriter can be very time-consuming, depending on the complexity of the graphics and whether or not fonts need to be downloaded or pages need to be reduced or enlarged. At present, the standard Apple system software does not spool LaserWriter output. Spooling allows a print file to be created and sent to the printer when it is available. In the meantime, the user is allowed to continue working on something else. Without print spooling, we have to wait and watch the status messages on our Mac screens until the printing is complete. If the printer is shared among many users and someone else is using the printer, this can be a very long wait.

Several options for LaserWriter print spooling have recently been announced. I know of only one, however, which is actually available now. LaserSpool from Mac America (and written in

France) provides an alternative to Apple's LaserWriter driver. Once placed in the system folder, a new icon appears in the Chooser window for spooled LaserWriter output. Clicking on the icon is all it takes to begin spooling. Actually, that's not entirely true - each time the LaserWriter is powered up, the spooler needs to be triggered by hitting Command-F the first time the LaserWriter is printed to. This is not a problem for us, however, because we leave our LaserWriters on all the time. Once the spooler is triggered, subsequent print requests are spooled. After clicking on OK in the LaserWriter dialog box, you see a message that says "Creating PostScript File" for a few seconds while the file is spooled locally (to either your hard disk or a floppy, wherever the system files are located). You are then free to continue with something else and the print file is sent to the LaserWriter while the computer is idle. Although it takes a little longer to get the page out of the printer, you more than make up for it with the time you gain using the Mac. You can cancel the spooling at any time by hitting Command-F again or by choosing the standard LaserWriter driver from the Chooser. A desk accessory also allows you to bring up the standard AppleTalk status message so you can check on the status of the LaserWriter you're printing to. I've only had this product for a few days and haven't tested it under all circumstances, but have not been completely satisfied with it. We have seen many circumstances where the PostScript file is created but is never spooled to the printer. When it does work, it is nice to have the computer free for other things, though. Other things I haven't been completely are that it isn't completely automatic (the Command-F requirement) and that there is no queue control (one cannot delete a file from the print queue or change the order of queued print jobs). Also, one cannot see a list of what is in the print queue.

Other products for LaserWriter print spooling have been announced but not yet released. Laser Serve from Infosphere (the folks that make MacServe) should have been released on Halloween. LaserServe is promising an impressive set of features, including all of the features missing from Laser Spool. LaserServe is installed as a desk accessory and the spooler can be turned on and off from the menu. The desk accessory allows full view and control of the print queue, letting the user cut and paste jobs in the queue. The spooler can be installed automatically at startup and can even recover jobs left in the queue. This means that if you turn off your Mac or it crashes (that never happens, does it?) with jobs left in the queue, they will be restarted when the Mac is. Also, if you turn the spooler off with print jobs in the queue, they will be printed when you restart the spooler. LaserServe uses a round-robin approach to sharing the printer among many computers. When a print job is completed on one machine, each of the other machines is allowed to print if they need to before another job from the original machine is printed. A number of other products for print spooling, both hardware and software, are supposedly just around the corner as well.

Mail Service

The only product I've seen for E-mail is called InBox, from Think Technologies in Bedford, MA. It is a very impressive system which offers all the features you could want in an E-mail system for your office. It requires a dedicated Mac to run the system and is rumored to put a lot of traffic on the network if you have a large number of users. I've not tested the product any further so won't comment any more, but if you need E-mail, check this one out. [Ed. Note -- A background version of InBox is now available, and a competing product, InterMail, is also out]

IBM PC Connections

Other than the Centram Systems board described above, one other IBM PC connection is available. Tangent Technologies provides an AppleTalk board for the IBM as well as a software package called PC Mac Bridge which is advertised to have many of the features of TOPS. Apple has also been promising an AppleTalk board for the IBM PC for some time now and rumor has it that they are nearing release of the board.



Products Mentioned

AppleLink, AppleTalk, HD20, HD20SC, ImageWriter, ImageWriter II, LaserWriter, LaserWriter Plus, MacDraw, Macintosh 512K, Macintosh Plus: Apple Computer, 20525 Mariani Ave., Cupertino, CA 95014 (408) 996-1010

DataFrame 20: SuperMac Technology, 1901 Old Middlefield Way, Mountain View, CA 94040 (415) 964-8884

dBASE Mac, dBASE III: Ashton-Tate, 20101 Hamilton Ave., Torrance, CA 90502-1319

EtherLink, EtherMac, 3Server: 3COM Corporation, 1365 Shorebird Way, P.O.Box 7390, Mountain View, CA 94039 (415) 961-9602

FastPath: Kinetics, Inc., P.O.Box 3341, Walnut Creek, CA 94598 (415) 947-0998

Helix VMS: Odesta Corporation, 4084 Commercial Ave., Northbrook, IL 60062 (312) 498-5615

HFS Locator Plus: PBI Software, Inc., 1111 Triton Dr., Suite 201, Foster City, CA 94404 (415) 349-8765

InBox: Think Technologies, Inc., 420 Bedford St., Lexington, Mass., 02173 (617) 863-5595

Interbridge: Hayes Microcomputer Products, P.O.Box 105203, Atlanta, GA 30348 (404) 449-8791

Laser Serve, MacServe: InfoSphere, Inc., 4730 SW Macadam Avenue, Portland, OR 97201 (503) 226-3620

LaserSpool: Mac America, 18032-C Lemon Dr., Yorba Linda, CA 92686 (714) 779-2922

Microsoft Excel, Microsoft Word: Microsoft Corp., 16011 NE 36th Way, Box 97017, Redmond, WA, 98073 (206) 882-8080

Netway 1000A: Tri-Data, 505 East Middlefield Road, Mountain View, CA 94043 (415) 969-3700

Northern Telecom, Nashville, TN

PC Mac Bridge: Tangent Technologies Ltd., 5720 Peachtree Parkway, Norcross, GA 30092 (404) 662-0366

PhoneNet: Farallon Computing, 1442A Walnut St. #64, Berkeley, CA 94709 (415) 849-2331

SpellsWell: Greene, Johnson, Inc., 15 Via Chualar, Monterey, CA 93940 (408) 375-2828

TOPS: Centram Systems West, Parker Plaza, Suite 220, 2560 Ninth St., Berkeley, CA 94710 (415) 549-5900

Michael Rogers is a Macintosh consultant specializing in networking, database applications, desktop publishing/graphics applications, and more. He can be reached at MWR Inc., 5930 Romany Rd., Oakland, CA 94618 (415) 658-4836



blocks/Hockabout



MACLINK-ing CP/M and MS-DOS (IBM) Documents Into a Mac And Back

Kenneth William Hirsch

After four and a half years of owning CP/M computers, I journeyed to the land of Macintosh with a load of baggage: 504 documents written as WordStar files on my old Osborne and Televideo machines. Before making the journey, I learned that one could transfer any data file from any computer to any other provided that both computers had some way of sending and receiving pulses over a telephone line to which they were connected simultaneously. Theoretically, each computer's modem and modem communication program would pull the data across, provided they were in the form of ASCII files, known as "Text" files in Mac lingo.

The Problem With ASCII

Changing the documents from many word processing, spreadsheet (financial and mathematical modeling) and data base (filing) applications into ASCII text documents shaves off the extra stored special command characters that make each document uniquely an outcome of its parent application. Wordstar-to-ASCII removes the pieces of information which indicate bold face, underlining and formats, and lots more. This loss of information is also true when converting MacWrite, Microsoft Word, Multiplan, Excel, and other documents into ASCII documents; the resulting plain "Text" file may be readable on other computers, but a great deal of the stripped information must then be recreated.

The wonderful *Spring '85 BMUG Newsletter* tutorial on "Transferring Files Between Computers" by Reese Jones (page 41) clarified a great deal for me. But, although many magazine articles and a couple of books on computer communication say it's "easy," I never could get the transfers to work right with my TeleVideo CP/M machine, which was the only computer I had available when I got my Mac+. I struggled for two weeks, with help from technically knowledgeable friends, to set up a simple and quick procedure to bring across that huge stack of documents. What ended up on the Mac required so much line-by-line re-working that it was a ghastly enterprise.

In the end, I paid about \$110 to get a copy of MacLink and the cable which would tie a Mac to an IBM. (Currently, the combination costs about \$90 from discount mail order firms, of which about \$25 is for the cable. The program can be ordered without the cable.)

What MacLink Does

MacLink makes document translations of several formats used on IBM and clones documents as closely as possible into the equivalent documents produced by applications used on the Mac, and vice-versa. In other words, it provides a simple and practical means of sharing information on disks among Mac and IBM users. No need to surround the Mac with another computer which duplicates an IBM PC and costs about as much (MacCharlie). No need get into the costs and complexities of plugging in a network (Appletalk).

Although there are a couple of other programs available to help execute such data transfers, what makes the MacLink program superior is the number of different conversions it can make to approximate a match between dissimilar applications and their documents. The possibilities are shown, below:



IBM to/From Mac

WordStar	--	MacWrite
Multimate	--	MacWrite
DCA (IBM revisable)	--	MacWrite
Microsoft Word	--	Word (CONVERT format)
Lotus (WKS or WRK)	--	Multipan (SYLK)
Lotus (WKS or WRK)	--	DIF
Lotus (WKS or WRK)	--	Excel (WKS)
Multipan	--	Multipan
DIF (Supercalc, etc)	--	DIF
DIF (Supercalc, etc)	--	Multipan (SYLK)
Text and binary	--	MacWrite
Text and binary	--	Text
Text and binary	--	Binary

Understand that the translations go as far as they can, where there is functional equivalency. Where there is none—for example, where the function NET PRESENT VALUE is included in the originating application program, but not in the receiving program, then there can be no translation of that particular information contained in the data document. However, the MacLink manual makes clear what functions and characteristics of the various programs do get translated and which do not.

The actual conversion and transfer procedure is quickly learnable and requires a minimum of technical knowledge. The manual is excellent.

Though the program is extremely easy to use, it is not automatic, for is it not possible to instruct the program to convert and transfer a batch at a time—i.e. all document files on a disk. One must be there to choose the type of conversion, then start each document transfer. Fortunately, MacLink beeps when it is through with a transfer, so it is possible to read, watch TV or whatever until the next beep signals completion of a transfer and readiness to begin another. For each transfer, the screen makes two requests for information:

- (first), What kind of file format is the original?
(User selects a choice from a menu.)
- (second), Do you want the translation/transfer to occur now?
(User clicks on a "go" button.)

CP/M (Osborne, Morrow, Televideo, etc.) To Mac and Back

Having already transferred WordStar documents from the Osborne's single-sided disks to the Televideo's double-sided disks by using Uniform (Media Master or Osmosis would have worked equally well), I again had to use the Uniform program to convert the documents to the PC-DOS/MSDOS format readable by an IBM machine's disk drives.

With my set of IBM disks full of data, a set of empty Mac disks, the cable, MacLink programs on Mac and IBM disks (system versions 2.1 or higher required), manual and Mac+, I cabled an IBM XT directly to my Mac+ so it could receive data at the 9600 Baud rate (960 characters per second). The IBM happened to have 640 K of RAM memory; both computers were equipped with two disc drives, which I find a great help, even though the MacLink manual says it will work with single drive systems.

Speed

Document data conversion and transfer is quicker if a direct cable is used. If transfers are accomplished via modem, they can go no faster than the capabilities of the modems and phone lines and MacLink—currently no more than 1200 Baud (120 characters per second). Transfers at 1200 baud would take eight times longer than direct cable connection transfers. Even so, I have found that direct cable transfers at the nominal 9600 Baud rate are actually slower than 960 characters per second.

Text and binary originals requiring no translation were transferred at perhaps 500 to 800 characters per second, with little or no delay. (They end up as text or binary files.)

WordStar to MacWrite transfers took place at about 2/5 that speed—say 200-320 characters per second. In other words, translation and conversion adds about one to one and a half times as much time to the process as is required for a straight text file transfer. Still, the time which MacLink saves because of its continuous operation is substantial when compared with proceeding by stripping a formatted document to make it into straight ASCII text, then transferring it to a Mac disc, then converting it to MacWrite or Word format, then reformatting and reconstituting it.

Outcome Flexibility

The MacWrite outcome documents converted from their WordStar origins present one problem which may or may not be characteristic of all MacLink-converted documents: they are difficult to rearrange with respect to internal reformatting. Typically, when a portion is selected, the black area indicative of selection is ragged along the side margins, conforming to the edge of the text rather than to the edge of the page.

I have found that MacWrite seems not to recognize the line of white space between "1. Do you have any system for finding out..." and "product or system you delivered?", where dragging the mouse to select seems not to have turned the background black. Trying to get rid of this space to contract the document seems impossible. If one persists in trying to select it, MacWrite says "You can change the header or footer only in its window". Umm???

It may be that there is some simple explanation for this idiosyncrasy, and some equally simple fix. If so, I haven't yet found it. But I have a "work-around".

Work-Around

Using Microsoft Word, convert the MacWrite document to a Word document. The line can be eliminated and there seems to be less difficulty with making changes, in general.

Evaluation

I liked talking with the proprietor of the company which produces MacLink. He was easy to phone, was genuinely helpful and avoided hype in describing what his program could do. Furthermore, the program worked when it arrived. Keep in mind that I have used it only with WordStar, ASCII Text and binary documents converted and transferred from IBM environment to MacWrite. Nevertheless, considering the large number of transfers performed smoothly and with ease, I believe MacLink to have given full value.



MacLink: IBM Data transfer to Mac

Harry Critchfield

The intention of the this article is to give the reader a brief understanding of the use of MacLink and the simplicity with which a transference of information from an IBM disk to a Mac disk, can be done.

The job which I took was to do a 12 page newsletter. The editor had used an IBM to compile the articles in WordStar. First thought was have someone retype the files on my Mac but this would take time away from me on the Mac. I had heard there was a connection between the two machines, and if true, I wanted to complete the project with just the Mac and myself. The device most often spoken of was MacLink - so I purchased it.

In the office I have both an IBM and a Mac and with the MacLink I was all set for the transfer. Within 3 minutes after opening the MacLink package I had the two machines cabled together (a cable comes with MacLink). One end plugs into the modem port of the Mac and the other end plugs into the communication port of the IBM. Continuing to follow directions, I transferred 50 pages of WordStar to the Mac within 7 minutes.

At this point in time, about 20 minutes had passed. Having completed the transfer, I read the Mac disk to see what the data looked like. There were little boxes at the beginning of every line of the WordStar transfer. Using search and replace of MacWrite I was able to eliminate these boxes. What I did was select a box, copy and paste it into the search area, and then placed a space bar mark in the replace area. After this I use PageMaker to complete the newsletter.

MacLink Software allows easy operation from the Mac for this transfer and your Mac controls the complete operation. The manual sets up a very clear understanding of how to perform the transfer operation. No more use of the IBM.

MacLink from Dataviz - list \$ 155.00.

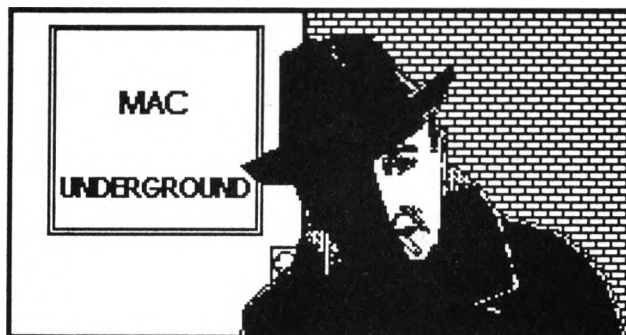
Msg. #1340 in **Public Messages** Posted on 05/29/86 at 09:36:05
To: ALL From: DAVID ROWLAND
Subject: HP12C
The HP12c calculator DA I have never did work. Now I cannot find it on the system. Was it always a bogey? Or does some version work properly? It was so pretty.

Msg. #1385 in **Public Messages** Posted on 06/01/86 at 16:20:33
To: DAVID ROWLAND From: SYSTEM OPERATOR
Subject: Reply To 'HP12C'
Dave -
The only version released into the public domain of HP-12c was a pre-release program. As soon as the author got his certified developer status with Apple, he took it back and sold it commercially through Dreams of the Phoenix. We can no longer distribute it.

-- Raines Cohen
SYSOP, BMUG BBS

The Mac Underground: What's there for me?

Copyright ©1986 Chris Halaska



The Mac Underground – hmm— sounds like some sort of software piracy ring or something. Or maybe it's a black market for memory upgrades. Actually, it's neither of those things, although you can find plenty of shady rumors floating around.

The Underground is many things, among them a source for the latest Apple gossip, a repository of the most useful public domain software around, and a purveyor of ground-breaking telecommunications software.

What is the Mac Underground anyway?

A long, long time ago— say, about a year before the Mac first came around — two people, Jerry Daniels and Mary Jane Mara, were thinking about the relationship between computers and people and how computers could be made most useful to everyone. At the time, they were getting paid to search online databases, struggling to keep up with the overwhelming flow of information. When they saw the Mac, they said, "Online databases are for the birds!" and proceeded to ditch those jobs and create Online Publishing Services (OPS), of which the Mac Underground (MU) is a subsidiary.



Jerry and Mary Jane had a vision: to make information retrieval as easy as using the Mac. Realizing that they couldn't compete with huge services like Dialog or CompuServe, they decided to start a small information resource which would provide Mac users with *timely* and



entertaining information about the Mac. Since this info would hopefully be useful in liberating Mac users from typical computer use, Jerry and Mary Jane adopted the spy motif of the French Resistance to place emphasis on the fact that they were providing 'intelligence' to the members of the Underground.

Underground Intelligence

The most intriguing part of the Underground is this intelligence, which is published in an online magazine format. Each weekday, articles called *Grapevines* are available. These are generally short (a page or two) and contain juicy tidbits such as the latest Apple rumor or a new product announcement. Believe me, their rumors are good - at least as good as what you'd hear at any given BMUG meeting—Jerry has his Apple contacts, too. Every weekend, an article entitled *On the Road* appears. The Roads are longer (4-5 pages) and consist of philosophical ramblings on the meaning of the Mac or on the importance of new technological advancements. Most often written by Jerry, many of these are of a truly visionary nature. Everyone who follows Apple knows that the release of the IIGS was a significant event, but leave it to Jerry to come up with some *really* impressive possibilities for this new computer.

By the way, since MU is committed to using the Mac to its full potential, many of these articles are spruced up with graphics. In fact, there are even weekly Underground Comix — *Tales from the Mac Underground*.

In addition, there is a weekly members' *Forum*, in which members' most common questions are answered and controversial opinions aired. There's also a bi-monthly *Feature* column, which presents notable people, for instance, Don Brown, the wizard behind MockPackage, or even the infamous John Dvorak. The list of intelligence goes on, but I don't often read the rest (reviews of products, comments on programming & desktop publishing, etc.), mainly because I don't have the bucks to download all that stuff!

An aside about Jerry Daniels: He's the most visible persona of the Underground and gives it much of its character. Also, Jerry likes David Letterman. A lot. After reading a couple of his articles, you'll begin to see some similarities between the two. JD likes to make fun of things that he thinks are stupid, mostly Apple and large online services such as CompuServe ("CashWeServe" to JD). Although the points he brings up are serious, he truly does make them in fun.

MU's software section tends to be out of date and is far from complete. You might want to forgive them if you realize that this is not their main thrust. Besides, the software that they *do* have is the most useful of the public domain programs and shareware that's around, including ResEdit, MockPackage, Servant and a large collection of clip-art. As an added bonus, there are dossiers (read: well-written instructions) for most of the public domain software they have available. I found their dossier on ResEdit particularly helpful.

So, I hear you say, I can get ramblings on the Mac and a *complete* set of Mac software on CompuServe, Delphi, or geez! even for free on the BMUG BBS. What's so special about this Underground thing?

I thought you'd never ask.

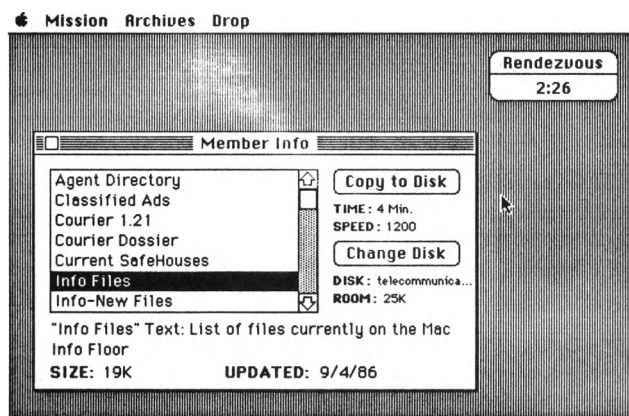
Underground Courier

You really needs to experience MU's *Courier* program to believe it. It works the way all telecommunications programs on the Mac should work—like Mac applications. Jerry wants

BMUG FALL 1986

you to be able to use your Mac with minimal effort and have a good time while you're at it. *Courier* lets that happen. I've never enjoyed using a program so much the first time—except maybe when I used the Underground's Microfilm Reader, but more on that later.

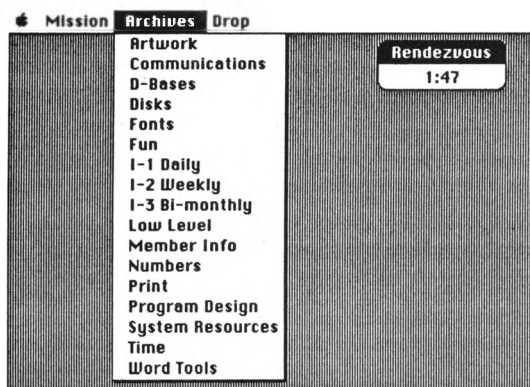
Initially, the mystique of logging on to the Underground was responsible for most of the excitement, but once I was connected, the pleasure came from using a program that allowed me to quickly select the types of files I wished to download from a Mac pull-down menu and then select the actual file for downloading from a Mac browser box. Quite a change for the better from using the text oriented BMUG BBS.



Even though *Courier* is good, the spread of the Underground has necessitated that they make it even better, and the next version, due out soon, will have other goodies to make it even easier to connect with MU, such as the ability to call the Underground when you're asleep and grab all the info you desire.

Underground Safehouse

Courier is only half of Mac Underground's system. It is not a stand alone telecommunications program but instead requires a host system with which to connect. The Underground's end is known as a Safehouse, and it contains all of the intelligence available to members.



A Safehouse is nothing more than a Mac, a hard disk drive, a modem, and the Underground Safehouse program. Oh yes, and someone to continually keep the intelligence flowing, known as a Bishop. Currently, there are Safehouses in 12 states across the nation (two in California,

BMUG Fall 1986



four in Illinois), and more are being added. The closest Safehouse to BMUG is one located in the 408 area code.

A word about Safehouses: when MU started, Safehouses were places to go to get great prices on Mac hardware as well as information. Unfortunately, due to the high costs (and huge headaches) of operating that type of outfit, the Underground ceased to sell hardware several months ago. I've heard, however, that they give you a *great* price on a modem when you join, if you don't already have one.

Microfilm Reader

In the middle of August, the Underground began publishing their articles in a new, non-MacWrite format. Again, Jerry Daniels and his crew of code cutters came up with a wonderful program that implements a great idea enjoyably. While MacWrite is nice, it's designed to produce 8 1/2" x 11" pages, often making it difficult to read text on the screen. Microfilm Reader was designed from the outset to be an on-screen reader. In keeping with the Underground theme, you load rolls of microfilm (the MU articles) into the reader to read the latest intelligence. It's like playing with a toy! And it teaches you something to boot. Hard copy of the articles may be obtained in the form of a contact sheet.



One controversial feature of the reader is the fact that it disables editing features and screen shots. These measures were taken to insure that the copyrighted material published by MU doesn't get spread all over, as was happening with their MacWrite based intelligence. This copyright protection has had a positive result, causing artists who were afraid to send their work to the Underground before, for fear of its spread, to become much less hesitant.

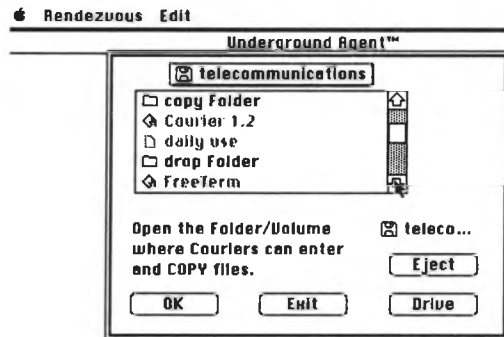
Underground Agent

Not wanting to let the Bishops have all the fun, a program has been developed which allows any member of the Underground (or anyone, for that matter) to have a simplified Safehouse. This



BMUG FALL 1986

program, called Agent, works by permitting the user to designate folders on his disk to act as drop-off and pick-up places for files.



This means that other people can call up an Agent using *Courier*, of course, and pick up or drop off files while the Agent is unattended. If the drop-off and pick-up folders are the same folder, people calling the Agent can pick up files that previous callers have dropped off. This is the least fun of the Underground programs. It lets you set up your pick-up and drop-off folders very easily, but then it becomes a text oriented program. That may be all that is necessary for unattended file transfer, but it's kind of boring.

The program is also a little buggy, sometimes mysteriously truncating files during set-up. In addition, Agent doesn't recognize more than one folder, so folders inside the pick-up folder aren't able to be seen by someone calling the Agent. This requires that all the files in the pick-up folder be at the same hierarchical level, preventing the person running Agent from separating them by subject. Supposedly, the next version of this program, Double Agent, will fix these problems and add more goodies.

How About a Little Pleasure?

Recently, MU opened up a new 'floor' in its Safehouses known as the Pleasure level. This section of each Safehouse includes such items as databases of film, book and record reviews, original fiction, and digitized art and sound, including animation and mild erotica. Access to this level is separate from access to the Information level, so I haven't yet had a chance to check it out. The word from the other members, though, is that it's pretty entertaining.

More on the Underground Philosophy

Maybe I should let Jerry Daniels tell it:

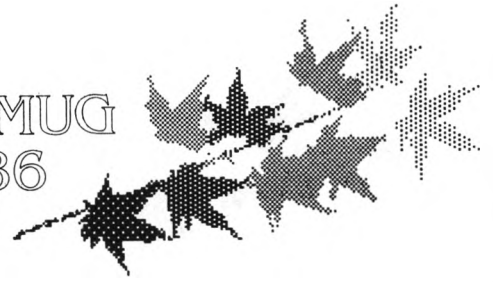
"Now we're getting down to the whole reason for the Underground.

"It's for the little guy. The Schmoe who can't type. The musician, the artist, the goof-ball, the misfits —> MY KINDA PEOPLE. We're just a Mac file transfer utility. We don't want to control the kinds of files people send. MU is for people who just want to use the Mac. You don't even have to know telecommunications."

And:

"I've tried to come up with metaphors that people like regardless of the outcome. MicroFilm Reader came about because I noticed that people like to play with their cameras. *Courier* happened when I noticed that no one ever tired of WWII spy novels or the movie *Casablanca*."

BMUG
Fall 1986



What more can I say?

What's It Gonna Cost Me?

The Underground doesn't charge online fees, instead, there's a yearly membership fee of \$59.95 for the Information level. I know that sounds like a lot, (to me at least) but lets look at the usage pattern of a typical member (me): This person logs on to the Underground for a half hour each week. Over a year, that's about 25 hours, which works out to an hourly fee of only \$2.40. Peanuts.

Agent is also \$59.95, but I think that this program is overpriced considering its limitations and problems. However, if you need something for simple, automated file transfer, this is it.

The Pleasure level is a bit more steep, checking in at \$79.95, but the fiction and graphics take more effort to produce than rumors and ramblings. Besides, we're talking quality entertainment here!

Now for the good news: as a Holiday Special, MU is offering a year's access to the Information floor (which includes *Courier* and Microfilm Reader), a year's access to the Pleasure floor, Underground Agent, and a Mac Underground poster for only \$99.95. If you add up the prices of each of those individually, you'll see that you're getting a half-price deal here. You can't beat that!

So Now What Do I Do?

The Mac Underground can be reached by phone at 515-472-9613 from 10AM to 4PM Central time, Monday through Friday or you can write them at:

Underground Headquarters
607 North Court St.
Fairfield, Iowa 52556

If you're still a little hesitant, demo copies of *Courier* will be available on the BMUG BBS. This will allow you to call any Safehouse from 4PM to 10AM, Monday through Friday and all weekend. The nearest Safehouse to us is down south at 408-374-0661. In demo mode, limited files are available, but you'll get a good idea of how *Courier* and the Safehouses work and a taste of the Underground. If demand is appropriate, I'll also try to make copies of Microfilm Reader and some articles available.

The Underground/Resistance theme may turn you off by making MU sound like a suspect organization. Believe me, I felt the same way at first. I sent them my money anyway, and from my point of view, I've received a great return on my investment. The Mac Underground is truly an alternative to the high costs of other online services and the long lead time of the Mac magazines. Most importantly to me, you're part of an organization that is truly interested in using computers effectively and efficiently. And not just today—tomorrow, too. Although the viewpoint of MU is somewhat biased, as it expands, it's going to need more and more people to write articles and draw pictures for it—you could influence the members of the Underground with *your* thoughts someday soon.

I'll see you at the secret rendezvous.

CompuServe Update: Dungeons and Dragons On-Line

Linda Custer

On Saturday, 14 June 1986, Ismelda was to be married. However, she had disappeared. Her father, Hophorus, was very concerned and called on a group of CompuServe users to get together on MAUG™ and find her. Chivas, Yow Meat, Gwydion non-Druitnain, Val-Daar, Gnillie, and Proteus responded to Hophorus's plea. After killing many Orcs and a dragon, meeting up with a mysterious cat, taking the cap off one very powerful bottle of water, and finding some gold and silver treasure to boot, the respondents returned Ismelda safely to her father. Of course, they were able to do this all over the phone in the comfort of their living rooms, courtesy of Macintosh, MAUG, and VMCO during an on-line game of Dungeons and Dragons.

The game

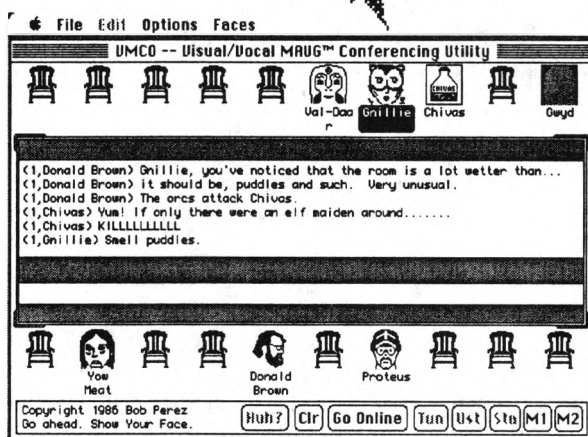
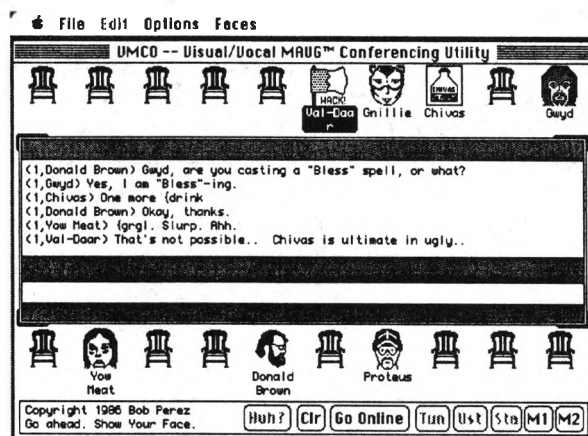
Dungeons and Dragons— or, more affectionately, D&D— is a fantasy role-playing game. Like all games, it has rules. However, D&D has no predefined gameboard or pieces. Players use their own imaginations to create personalized characters and a unique gameboard (which the players see or are told about only as they traverse it), so every D&D game is different from all others.

Before a D&D game begins, the DungeonMaster (DM) consults the D&D rules (available from TSR) and maps out his own unique "world" for the game. Usually, the world contains castles with many levels, forests, villages, moats, fair maidens, knights, elves, dwarves, orcs, buried treasures, and dragons, and is much like the world that J.R.R. Tolkein created in *The Hobbit* and the *Lord of the Rings* trilogy. The DM then gathers together some friends, each of whom takes on a character of their choosing, and the game begins. Although the players and the DM have much free choice in how the game progresses, many actions are dictated by the roll of various dice. For example, the personality and strength of players, the amount of damage inflicted or sustained during battle, and the outcome of certain events are all decided by a roll of the dice.

The VMCO advantage

D&D is usually played by a group of people together in the same room. Since Bob Perez's VMCO (Voice/Visual MAUG Conferencing) program allows people to "see" each other's icons and hear each other's words, a group of MAUGgers decided that a D&D game would make a fun conference. I watched the game they played on 14 June from the sidelines and was impressed with the fun all the players had.

Before the game, each player took pains to create a set of icons for their characters' faces and also in some cases for their weapons! The icons are beautiful and very descriptive, as you can see from the screen shots shown below. During the game, the players used simple VMCO commands to change the face on all players' screens to suit the mood of the moment. Then, when everyone assembled, the DungeonMaster announced that the adventure started at a wedding party. As soon as the players heard of Ismelda's bad fortune, they followed a tunnel to trouble and treasure and finally Ismelda. An audience assembled and looked on in amazement but courteously refrained from commenting during the game. All in all, this conference was a very different and enjoyable way to pass a few hours, and it "came to life" because of VMCO.

[illegible]

321

2400 Baud Modems

Timothy Standing

Any one who hasn't yet bought a modem, should seriously consider buying a 2400 baud modem. I use my Mac both for logging onto the BMUG bulletin board and for working on a VAX from home. Using a 2400 baud modem means that downloads go twice as fast, and that editors that you use on a mainframe actually act normally. I guess I don't have to persuade any one that 2400 baud is a good thing. Which one and how much?

I bought a U.S. Robotics Courier 2400 Modem a year ago, and have had nothing but great experience with it. It seems to be immune to all but the worst phone line noise. I have called mainframes in Boston and Washington using MCI and have never had any problems. This modem is also fully Hayes compatible, including all the registers. It has some great features like on line help, busy signal detection (no more waiting for your modem to time out) and dial tone detection (for those of use who use MCI). The best thing is that it comes with a summary of all the instructions printed on the bottom of the modem (no more searching around for the manual when you are trying to send a file to a friend). You can now get one of these for \$329 from Mail Order places like TeleMart (800-622-6659). (I usually get things within a week from TeleMart and their prices are great).

Other modems to consider are the 2400 baud modems from CTS. These come in Hayes compatible and non-Hayes compatible flavors, with about a \$40 price difference between them, the Hayes being more expensive. These modems lack a speaker, but otherwise appear to implement all the features of more expensive modems. They are also just about the cheapest around; when purchased in bulk they can be gotten for \$234, including shipping and tax. Cables are not included. Long term reliability cannot be confirmed since they've only been on the market a relatively short time, but few bad reports have come in.

Msg. #2077 in **Public Messages** Posted on 07/10/86 at 15:59:17

To: ALL From: JIM TAKATSUKA

Subject: RE: Big Mac Screen

A company down in Canoga Park called MicroGraphic Images produces a 1024x1024 Macintosh screen that maintains the 72 dpi resolution of the Mac. This isn't just a video adapter and a large monitor (a'la the Mentaureis/Conrac combo) which simply enlarges everything. The MegaScreen gives the Mac a full 1024x1024 display (vs. the current 342x512). Smarter software that doesn't assume a fixed screen size can take advantage of the extra display area (e.g. allow you to see two full pages at the same time). \

Monitor and daughter board combo also allows output of NTSC signal at standard TV resolution. It also has the ability to display an NTSC image on the 1024x 1024 screen using gray scales. The whole setup goes for \$2995 retail, which isn't too bad considering devices which only output an NTSC signal go for almost that much.

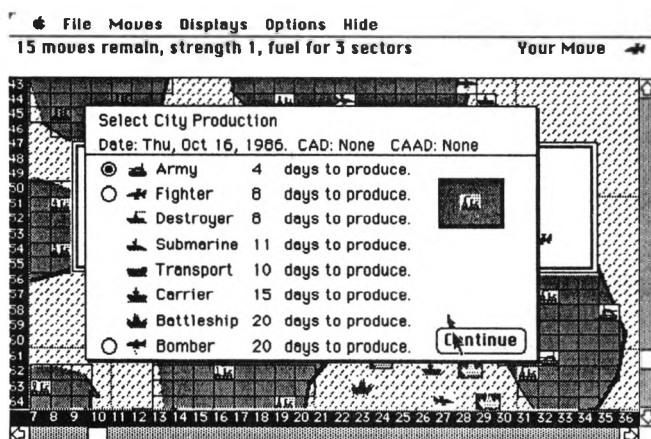


Strategic Conquest Plus

Steven Fransen

Sure, PageMaker is good. FullPaint is fine, etc. But now we have the ultimate Mac program, the Strategic Conquest Plus. It comes on two 400k disks, one of which just contains digitized sounds. When booting up the program, you have several playing modes: (1) single player, (2) two players on one Mac, or (3) two players on two Macs using Apple talk. For the dedicated: the world grid is 70x70 vs the old version's 60x100. Between the two versions, the production Levels are also different, see the following list:

	OLD	NEW SINGLE PLAYER	NEW TWO PLAYER
army	4	4	2
fighter	10	8	4
destroyer	15	8	4
submarine	15	11	5
transport	15	10	5
carrier	25	15	8
battleship	25	20	10
bomber	20*	20*	10*

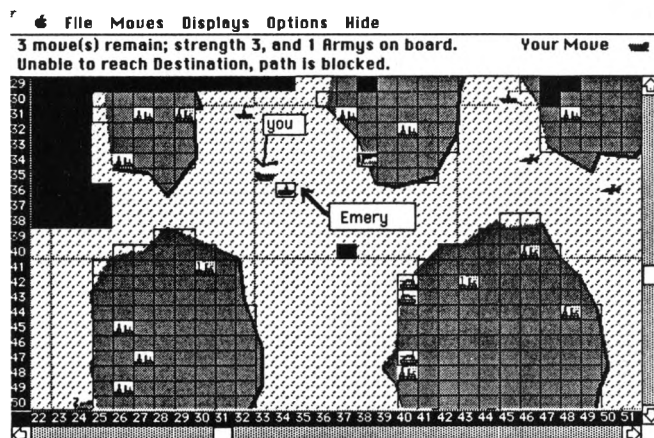


The world is smaller, islands are closer together, and you usually make contact with the enemy much sooner than in the older version. The most cities you will ever find on any given island is 5. The look of the islands and ocean is more realistic. There is an option to turn the grid on/off which can take a while to do, like 30 seconds. You can now pass on a given piece and come back later, which is very useful for transport and tanks, so you can pick up a tank and then drop them off on different islands. Also, along with city destination, you now have a idea called army destination. This will automatically send your tanks to a pacific city so you don't have to move them - the computer will move it for you. This is useful for cities next to a port city or a long way from the coast. The feel of the new version is much nicer. The sounds are digitized and a big improvement. The game is much more pleasant to play and look at.

The biggest difference is in the STRATEGY. Peter Merrill must have spent a lot of time re-doing his algorithms. If you played the old version much you soon realized that battleships were the key to victory. This is no longer the case. In fact, a lot of strategic re-thinking is necessary or you will lose badly. The destroyer, the submarine, is very powerful and is the main key to this

BMUG FALL 1986

version of the game. The fighter, the battleship, is not as powerful, as it was before, but it makes the game more playable against the computer. The bottom line is that Peter has better integrated all the different pieces so they are a lot more useful and necessary. The computer plays much smarter now too. So watch out! For example, in the old version, when the enemy discovers the presence of a battleship, it would throw every fighter it has at you in a vain suicide attempt to sink the ship. Now, it notes the battleship's location and instead of attacking with fighters, it may send a submarine to finish you off. And, yes, subs can now sink battleships with two shots!!



As to difficulty levels: I had regularly played at level 15 before, and after getting completely overwhelmed four straight times with the new version I moved down to level 11. It takes a while to get used to the new strategy required to beat the computer. I'm at level 14 now and struggling.

The TWO-PLAYER MODE is a different animal altogether. Production levels are halved, and the number of sector moves per turn is doubled for all pieces except the fighters and bombers. This is necessary to speed up the game. You will find that your opponent takes a hell of a time to complete his turn, so get comfortable and dig yourself in for a long battle. But, this mode is an incredible amount of fun. (Note: Apple talk cables are not required - printer cables work just as well). The anticipation of the first contact is very very intense. It provides you with a wonderful opportunity to be totally ruthless, to be totally merciless, to maim and hurt those close to you, at least in a metaphysical sense. There is a funny option which allows you to "Send Messages" to each other such as "I hope your crew has lowered the lifeboats, as your battleship is about to take two torpedo's at mid ship). PLEASE, BOTH PLAYERS, DO NOT START TYPING TOO FAST as it will hang the program up until it catches up to your typing. You may also invoke a "Talking" option, and the good old MacInTalk voice will keep you informed as to the progress of the game. The sound is so good that you will want to hook up your stereo so you can really enjoy it! There's a "Battle Summary" feature which provides you with a list of the battle activity after each enemy 's move. These last two options, however, will slow things down considerably. There is a Auto play feature to start the game up to see what the computer would do so you can learn from his moves.

Password protection is used so that the other player can not cheat and look at your side of the world. The game will play against the computer with one player or two players on one machine, or two players on two machines. The two players on one machine is slow on a 512k Mac. It is better on a 1 meg or more Mac. I have played it with a prodigy: it was OK but not really any faster. WARNING: try not to forget at all times that this is only a game, or friendships would be

BMUG
Fall 1986



strained. In our first battle, my friend realized, by day 35, my quest for world domination was more than just a vision. He began whining about how the game was buggy and how his pieces weren't moving properly, and then, as I was closing in for the kill, without warning he abruptly QUIT. I guess this game brings out the worst in people!

However, the main weakness is in the random generation of the world before each game. There is not quite the element of surprise as one finds in the old version. As a rule, the islands are uniformly smaller and there are never more than five cities per island. No such pattern existed in the older version, as you might find islands with 10-12-15 cities, huge interconnecting islands, or useless islands. Perhaps Peter changed the random generator in response to complaints that it took too long to finish a game. I just wish he hadn't messed with that particular aspect of the program. However, without question this is STILL the most entertaining game written for the Mac. This new version is more than a cosmetic upgrade to the older version. Because of the different strategy required to beat the computer, you're basically getting a different game. The new version lists for \$59.00. For an upgrade, send in your master disk plus \$15. I would try convince PBI to let you keep your master, as you might want to occasionally go back and play the old strategy. The two player mode truly makes this a unique experience. For those who do not live close to other strategic conquest maniacs, PBI will be tweaking this new version shortly -should be available by 10-28-86 - and they are definitely considering adding modem to modem play capability. (Rumor has it that they want to purchase as much AT&T stock as possible before they release it).

Buy it, you'll like it. I played the new version for about 2 weeks now, and the game is very different then the older version. The strategy is much different! The object is to locate other islands much faster and get transports there much sooner than before. The key is good RECONNAISSANCE, not battleships like before. The idea is to get a good supply line of transports going and support it, and to sever the enemy's supply lines. The universe is smaller, but the graphics and sound are DRASTICALLY improved! There are far more islands and they all each have up to 5 cities every time! As the islands are now closer together, your fighter, as he flies out into darkness, finds another island almost every time. The authors deserve our support as it is way better than any other game out there. All in all I think this is the best game on the Market ! PBI has got a WINNER here. Good luck with the game.

Strategic Conquest II \$59.00
PBI Software
1111 Triton Drive
Foster City, CA 94404
(415) 349-8765



MCS: Multichannel Communications System

Paul B. Blood

I investigated many programs while looking for a communications package for use in transmitting files, data, and programs used in research. One of the major requirements was to be able to transfer information to and from Norway, where my sister, also a Macintosh owner, is currently living. We needed a reliable and cost-effective method that would allow for the timely exchange of information.

We had tried ordinary telephone calls supplemented by mailing disks. This became very expensive and still did not give satisfactory results. We had to wait for the mail to transfer any program or formatting changes. Delivery time to Europe via air mail can vary from as little 4 or 5 days to as much as 6 weeks, and the disks may not arrive in usable condition.

I believed that some method of electronic mail was what we needed to use. I investigated the services of Telenet, Tymnet, Bitnet and many other suggested services. The major drawback to all of these services was that to use them we had to move out of the Mac environment and into a Text Only environment. I was told that the sacrifice of formatting and graphics was the trade-off for getting our information back and forth within a predictable time frame. This was unacceptable.

This was the dilemma when I had the good fortune to receive a program called MCS as part of the SPRING '86 *BMUG Newsletter*. MCS, the Multichannel Communications System, was developed by Yves Lempereur for in-house use at Mainstay, and is being distributed by him as shareware (suggested contribution \$10). Mainstay, developers of *MacBooster*, *Telescope*, *TypeNow*, *PacPaint*, and *BINHEX*, had already faced a problem similar to ours. They had to make transfers to and from Europe via a satellite link which was noisy, expensive, and had a long return time. Mainstay found that XMODEM file transfers were not effective because of these problems. They developed and implemented in MCS their own proprietary protocol, X.MS, similar to the CCITT X.25 protocol. It eliminates any noise and the problem of the long return time. MCS uses, in addition, a higher level protocol which allows simultaneous upload, download and chat by keyboard using different channels. Upload, download and chat all take place in the same length of time as a single XMODEM transfer, making MCS very cost effective. The MacBinary format is used for file transfers, so any kind of Macintosh application or document can be transferred.

As suggested in its accompanying documentation, MCS is most easily used if viewed as an upload-only program. Two persons who wish to communicate must both be using MCS. Each selects a series of files from a dialogue box. One operator selects **Dial** from the File menu, and then, if a connection is made, each of the two Macs running MCS simultaneously uploads a series of files to the other. The uploads will be done in the order that the files are selected. The MCS operators can chat during the file transfer by entering lines from the keyboard, which can be edited in the usual manner prior to pressing the Return or Enter keys. A line can be 80 characters long and can contain any Macintosh character.

MCS has the basic user friendliness of any Macintosh application. However since it was developed to meet Mainstay's in-house needs, it has some rough edges. Though all transfers are saved to disk, MCS does not check to see if there is adequate space on the target disk. If a file is received with the same name as an existing file, the pre-existing file is overwritten. To change the communications parameters or the phone number to be dialed you must use the companion

BMUG
Fall 1986



program *MCS Edtt*. No printing of the file being transferred can be accomplished during transmission, and the only way to print those items transferred during the "chat", is to complete the call, "Hang Up" and then use Command-Shift-4 to print the window.

We have been using *MCS* over the regular AT&T voice lines for some time now, with great success. We have found the cost to be near, but just slightly higher, than the cost of sending the same material by Air Mail. Our average cost for sending approximately 10K in each direction has been \$1.81.

We had some problems in the early stages of using *MCS*. We contacted Yves Lempereur at Mainstay and were immediately given the help we needed to set up the program, making it now one of the easiest to use and most cost-effective that we have. We find it not only valuable, but indispensable.

I want to thank Mainstay for allowing Yves to offer this program as shareware and to support it as a public service. It is available through most BBS's, on BMUG Disk #40, and can be downloaded from CompuServe *et al*.

Msg. #4168 in **Public Messages** Posted on 10/30/86 at 15:36:16

To: ALL From: MARK ACE

Subject: MS Word Questions

Is there a utility for recovering those MS Word files that are listed as MW0000, MW0001, etc.? Another question: lately, the screen has not been keeping up with my tpying, in both Word and Pagemaker. It's not that my tpying is so blindingly fast, but the words that appear on the screen often miss some of the characters I've typed. There is no such problem with MacWrite. Is this delay between typeing and display on the screen just a characteristic of Word and Pagemaker??

Msg. #4193 in **Public Messages** Posted on 10/31/86 at 22:59:15

To: MARK ACE From: RAINES COHEN

Subject: Reply To 'MS Word Questions'

Mark -

I don't know of any utility to recover MWnnnn files, but the usual suggested technique is to change their type to TEXT and try opening them in MS-Word (rename them first!) If that fails, call Steve Costa, BMUG's file recovery expert. I've seen a program that will convert MS-Word files to TEXT, but it is not yet out (it will probably be ShareWare).

The "lagging" of typing you refer to is probably a generic Mac problem, as only 14 "events" are stored (usually) at any given time on the Mac. Events include window updates, mouseclicks, and keypresses and keyreleases. If the program doesn't call "GetNextEvent" fast enough, it can miss some characters. Try making the window smaller, and working in a small, simple, installed font, size, and style. Also try turning off AppleTalk, disconnecting things from the serial ports, and typing in another program, if you're desperate.

-- Raines Cohen

SYSOP, BMUG BBS

Two for the Lab: *MacADIOS* and *LabVIEW*

Paul F. Daley

Macintosh users are accustomed to breakthrough products. Examples of entirely original Mac software concepts abound; think of *MacPaint*, Telos' *Filevision*, or *MacSpin*. Software available in other computing environments has achieved new levels of accessibility. On the Mac, user interface standards are realities, and "data exchange formats" are transparent to the user who is trying to get a job done. These are breakthroughs that add power to all the different products.

Scientists and engineers have flocked to the Macintosh. For all but the professional theoretician or programmer, no one class of software satisfies any but a subset of real computing needs. The Mac permits users to hand information between packages with unequalled ease. The capacity to easily use a number of programs together puts the scientist back in charge of all aspects of his ideas: from hypothesis to publication. Data can be acquired, analyzed (by means of statistical packages, or by comparison with simulation output), graphically presented, and coordinated with text for presentation.

Data analysis, with tools such as *Excel*, *Cricket Graph* and *MacSpin*, has never been easier. Presentation capabilities are well known to most Mac users, as the integration of graphics and text is one of the hallmarks of Mac output. However, the first step in much scientific work, data acquisition, requires more specialized tools.

Two products were released this year which address data acquisition in the best of Macintosh style: *MacADIOS* from GW Instruments of Cambridge, Massachusetts, and *LabVIEW*, from National Instruments of Austin, Texas. Both are high quality products which allow the Macintosh to shake hands with the physical world, measure real parameters, and control experimental apparatus. They are breakthrough products. Whereas previous data acquisition products on the Apple II or IBM PC usually required specialized programming skills (and even sophisticated hardware knowledge), these allow the researcher to directly address his own programming needs.

The two packages manage the Mac's contact with analog reality in different ways, and really need to be treated independently. *MacADIOS* is a specific piece of hardware with matched software. Subroutines specifically optimized to control this hardware and manipulate the data can be called from relatively simple programs. *LabVIEW* is a software package for controlling data acquisition: hardware accessories are available but not bundled. You program in *LabVIEW* through the creation of a "wiring diagram" showing the flow of data; *LabVIEW* executes the diagram as a program.

***MacADIOS* : All you've got to have...**

MacADIOS (for Macintosh Analog to Digital Input/Output System) is a self-contained hardware and software subsystem for data capture and instrument control. It is the brainchild of Glen Weinreb, who pioneered the circuitry and software concepts as part of an electronics course term project. Seeing many existing products in the Apple II and IBM PC market, Glen jumped into the Mac world, where unparalleled graphics and processing speed would distinguish his progeny from the rest of the pack.

The *MacADIOS* hardware is a solidly built electronics chassis, with about the footprint and thickness of an HD-20 hard disk. The front panel is divided from left to right into four panes:



indicator lamps, timing connectors, digital I/O, and analog I/O connections. There are eight analog input channels; four analog outputs; and sixteen digital inputs and outputs. (Note that a digital "input" or "output" corresponds to a bit, each of which can be part of a digital "byte," or a switch for control purposes). All can be connected with jacks for use in prototyping (BNC for the analog channels, and "bannana" plugs for the digital) or with 37 pin "D" connectors to attach the *MacADIOS* to more established instrument setups with ribbon cables. Data is transferred to the Mac through the modem port, placing a ceiling of about 22K samples per second on data throughput.

The electronics chassis is really a micro unto itself, driven by a 6502 microprocessor. The Mac sends commands to the 6502, which in turn controls the analog-to-digital and digital-to-analog converters, and the digital I/O. Additionally, there is a clock which can be used to synchronize the *MacADIOS* with external events, or internally to control sampling.

The bundled software enables you to control the *MacADIOS* in a number of ways. It includes a general purpose data acquisition package written in compiled C, the *MacADIOS Manager*; two specialized packages written in Microsoft Basic (2.1), called the *XY MACorder* and the *Mac Speech Lab*, and two libraries of assembly language routines, one addressable from Microsoft Basic, the other from Aztec C. The *MacADIOS Manager*, the *XY MACorder*, and the *Mac Speech Lab* allow you to get "up and running" almost immediately, though for specific experiments unique programs will almost always be created by the user with the over 50 control and number crunching routines available in the libraries.

The *MacADIOS Manager* is an array-oriented package: space is set aside in memory (a "waveform" in their jargon) and then filled by an input channel. A flexible "experiment" window allows the user to define operations using much the same command structure as the Basic libraries. Instruments can be controlled, data taken, and the resulting "waves" displayed on screen (Fig 1) or saved to disk in a compact binary format. The *Manager* lets you see how the *MacADIOS* operates without having to generate a lot of code, yet, by giving you experience with the syntax of the instrument it helps you get started with your own custom applications. Though the *Manager's* disk files can not be directly opened by packages such as *Excel*, simple Basic programs can be written to open them and write them back to disk in ASCII.

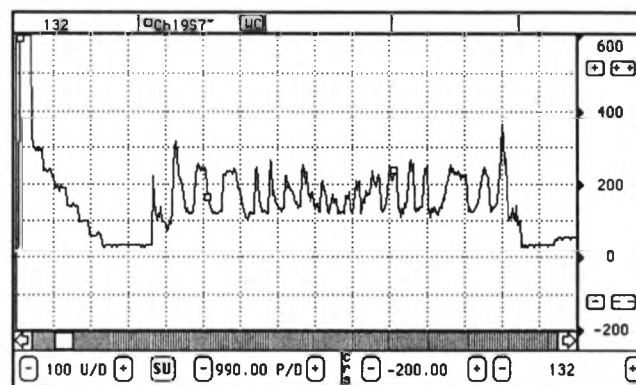


Figure 1: Tree ring X-Ray digitized with a scanning photometer at UC Berkeley's wood products lab using the *MacADIOSManager*... scanning time about 20 minutes. Center of diagram approximately corresponds to center of cross-section of trunk; peaks are harder late year growth.

BMUG FALL 1986

The *XY MACorder* and the *Mac Speech Lab* are full-blown applications and powerful instruments in their own right. The *MACorder* emulates X-Y plotters (where an analog input or time is the X axis, and a second analog input is the Y axis), and uses low-speed analog input routines, 10 samples/sec or less. In contrast, the *Mac Speech Lab* is a high-speed single channel instrument capable of accepting sound data at the full speed of the *MacADIOS*. *Speech Lab* can produce sound spectrograms of a quality not matched by instruments costing many times the price of *MacADIOS*. A sampler of further high-speed routines, called *5 Instruments*, is also included. (Fig 2).

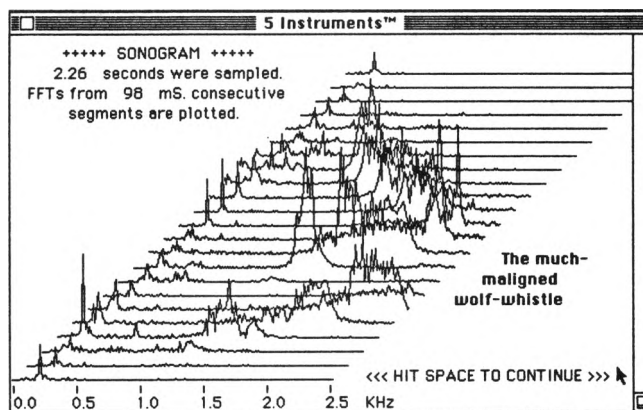


Figure 2: Digitized sound

Since these applications are included as Basic files, they act as extensive tutorials for using *MacADIOS* by means of Basic and the assembly language libraries. The libraries are replete with routines which perform such operations as oscilloscope emulation, Fourier transformations, and other signal processing operations. Internal documentation in the example programs' source code is very good. The user who is not principally a programmer will find that the existence of these routines and their access from Basic makes generation of new applications fairly straightforward.

LabVIEW: And now for something completely different.

National Instruments has taken a quite different approach to linking the Mac with the lab world. *LabVIEW* stands for *Laboratory Virtual Instrument Engineering Workstation*. *LabVIEW* generates virtual instruments (VI's), which, like real-world instruments, have front panels with controls the user operates (Fig 3), and various styles of output. In the back of the virtual front panel, a "wiring diagram" specifies the instrument's algorithms and operations (Fig 4). How is this distinctive? *The diagram is the executable code!* Arithmetic operators, program control structures (for-next, while loops, etc.), file I/O, and other familiar programming devices are represented as icons, and data paths are specified with drawing tools (a spool of "wire" to interconnect operators). The paradigm is similar to the data-flow concept used in *Stella* and some mainframe experimental programming languages, but in *LabVIEW* the concept is fully developed, intrinsically parallel in operation, and hierarchical.

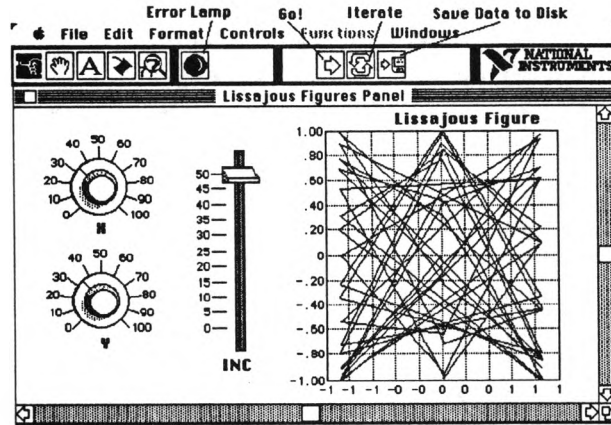


Figure 3: A *LabVIEW* Virtual Instrument front panel... those knobs turn, too!!

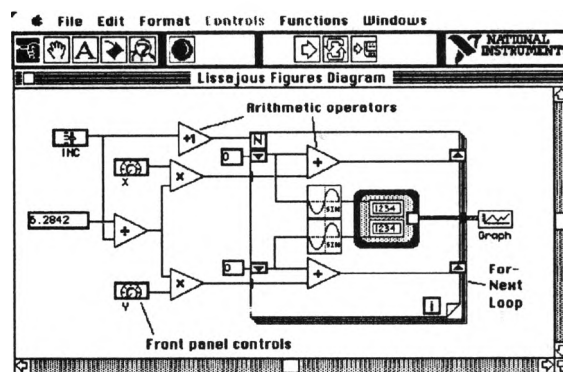


Figure 4: Diagram for the Lissajous example shown in Fig 3. *LabVIEW* connects to instruments through the Hewlett-Packard Interface Bus (HP-IB), or IEEE-488, a parallel interfacing standard well known in electronics testing labs.

If, for instance, you are starting with someone else's code, the whole structure of the *LabVIEW* programming language makes it easy to see what's going on. The overall flow of data is explicit, even obvious; with a conventional set-up you must read twenty, thirty, even fifty pages of code. Hiding particular details in custom icons makes the overall flow visible. Once a real-world HP-IB instrument's control is worked out (usually a matter of determining its address on the HP-IB bus, establishing "hand-shaking" with the computer, then issuing ASCII commands to specify functions, ranges, etc.), the virtual instrument controlling it can be reduced to a custom icon with a connector "pinout" to establish connections in other, higher-level virtual instruments. Even lower level operations such as initializing the bus and sending commands are programmed by graphical controls. The resulting diagrams are self-documenting: Details are hidden but available, and the flow of events and operations resembles familiar flow-charting techniques.

The *LabVIEW* interpreter has reasonably good speed at the moment: a bit slower than a good Basic. It checks in-line for syntax and for type mismatch in data, in the manner of MacPascal.



BMUG FALL 1986

As with the *MacADIOS*, a full set of pre-compiled "VI's" come with *LabVIEW* which perform complex operations: initializing the HP-IB, signal conditioning, creating graphical displays, statistical manipulation, and the like. Those familiar with the IBM-PC package *ASYST* will find virtually all of its functionality present in *LabVIEW* (and without the need to learn a custom Forth-like language).

LabVIEW can address instruments through the RS-422 serial ports directly. Connection to the HP-IB bus requires an additional translation box, a modem-sized unit which connects through the Mac's modem port. Data transfer rates with the translation box are usually limited by the instruments sitting on the bus, but can be as high as 57K BPS. Faster data acquisition will be possible by way of the MacBUS, an SCSI expansion chassis which will accept 16 bit PC-AT style data acquisition cards for data throughput at speeds near 1,000K BPS...real quick. It is to be delivered by National Instruments this fall. In any case, those interested in HP-IB must also purchase the devices that actually convert physical information into data that can be sent to *LabVIEW*.

Comparing the Two Products

Both of these products are powerful in their own ways. The software for the *MacADIOS* is very high performance, but specific to their hardware. It can be used easily with the kind of programming skills to which people are customarily first introduced. At the present time the *MacADIOS* would be easier to use for very fast data acquisition, for instance to take speech spectrograms and break down the sound into its component waveforms.

LabVIEW is more general purpose and open ended. It allows fluent interaction with a class of hardware devices, and can also be used for simulation. With its entirely graphical interface and intrinsic object orientation, *LabVIEW* code can be read, understood, and manipulated with relative ease. The ability to get an overview of what someone else's program is doing makes collaboration easier, assistance possible. With the next generation of hardware from Apple, *LabVIEW* will make possible a very simple and elegant manipulation of very powerful capabilities indeed.

PriceTags & Users' Needs

At first inspection, both of these routes to automated data acquisition seem to come with rather stiff price tags. *MacADIOS* retails at \$2500 (hardware + software). *LabVIEW* comes in at \$1900, though the price is not firm at this writing. For the GP-IB Mac [Mac-to-HP-IB translator box] add about \$500; if you're really hungry for fast data acquisition add ~\$1400 for the MacBUS. These prices will send shivers up the spines of prospective users. However, this particular marketplace is notable for fancy pricetags. Comparable data acquisition subsystems in the IBM world (Keithly, Isaac/Cyborg, etc.) fall into the same range, and *ASYST*, the closest software package for the IBM comparable to *LabVIEW* retails for over \$2200!

Furthermore, it's hard to overemphasize the value of the accelerated movement down the learning curve one achieves in using either of these products. I've spent too many dreary hours interfacing instruments with computers...with both the *MacADIOS* and *LabVIEW* I've cut that time to a tiny fraction, and been able to get down to the business of getting data. Weeks spent trying to work your way to the point of reasonable (or even of initial) productivity translate into LOTS of lost dollars and lost opportunities to make progress with the problems at hand.

BMUG
Fall 1986



I would hope that more easily usable products like these will engender some healthy competition and changes in pricing and availability, to make these tools standard and especially to put them into the hands of student users. Automated experiment control and data capture both enhance experimental quality, and increase the ability of the scientist to concentrate on real-world phenomena, which is what we're here for!

MacADIOS
GW Instruments
3 Ames Street
Cambridge, Mass, 02139
(617) 577-1524

LabVIEW
National Instruments
12109 Technology Blvd.
Austin, Texas, 78727
(512) 250-9119
(800) 531-4742
(800) IEEE-488 in Texas (!?)

, Lawrence Livermore Lab
Environmental Sciences Division



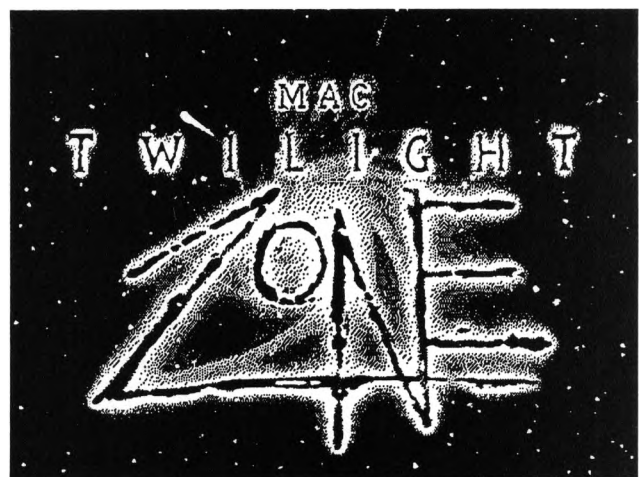
BMUG FALL 1986

Three of the Start-Up screens
from BMUG Disk#46



Author Unknown

Scanned by Phil Sue



Scanned By Phil Sue

BMUG
Fall 1986



Business

BUSINESS

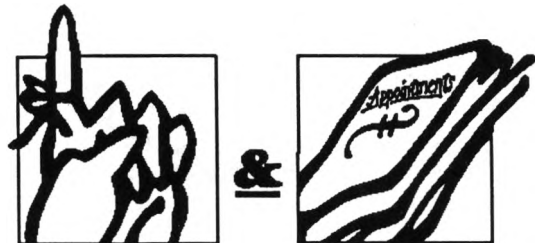
You Don't Need \$250,000! • Publishing Mac software successfully	Shepard Tamler	335
A Review of the Excellent Exchange • A collection of Excel applications	Don Scellato	338
Microcomputer Consultanting • Pointers and pitfalls in working for yourself	Audrey D. Cole	341
Big Guns: Double Helix vs. Omnis 3 Plus • Comparing the two database systems	Rick Eisner	343
dBase Mac and dMac III • Comparing the Mac and IBM database systems	M. Joseph Schaller	346
Record Holder • A simple and cheap database system	David Morgenstern	349
Poor Tom • An associate database	Glenn Harris	351
Macintosh Statistics Software • Six programs for statistical analysis	Michael La Belle & Melissa Austin	353



You Don't Need \$250,000! Shep Tamler

It is true that publishing a Mac software product doesn't mean instant access to Easy Street. But it can be done successfully, and may not be as difficult as it's sometimes portrayed. According to the article "So You Want to Be a Software Developer", in the August 1986 issue of *MacWorld*, "creation of a moderately sophisticated package - production, manual writing, initial marketing, and so on - costs at least \$150,000 to \$250,000. Some say twice that." I SAY PHOOEY! You don't need a quarter of a million dollars to profitably release a modest Mac product!

IMAGINE
SOFTWARE



My company, Imagine Software, has released a successful product, *Smart Alarms and the Appointment Diary*, for about one fifth the cost of the lowest number given above. The Editor of *MACazine* has said that "*Smart Alarms* is one of the most useful desk accessories ever invented". *MacUser* has given us five mice, their highest possible rating. It's selling well. I don't expect to become conspicuously rich from it, but I am making my car payments. And we're doing better than our projections.

Imagine Software certainly never started with anything close to those kinds of big bucks. But we did have some other things. You, as BMUG members, probably have access to these as well. Let me give you an idea of what we've done.

First and foremost, to put out successful Mac software, you'll need an intimate knowledge of the Macintosh interface. Before starting Imagine I'd been involved with the East Bay Mac Users Group for almost two years, so I had a real sense of what "good Mac programs" were like. I'd spent countless hours discussing the impressive features in the latest hot programs, and equal time thinking up improvements for programs that could use them. This is a hobby many Mac fanatics share. It pays off in spades in the Mac publishing business.

About a year ago, I first saw earlier Australian versions of *Smart Alarms and the Appointment Diary* during the MacWorld Expo in San Francisco. I could tell right away they had exceptional promise, but I also could instantly see a few things that needed to be improved. Although I'm not a programmer, I was able to write up a list of features that I felt needed to be added or changed before I'd agree to work with the product - and the current release has nearly every one of these.

In addition to recommending actual programming changes, my knowledge of the Mac interface enabled me to take the Resource Editor and improve lots of little things - like the text the user is presented with, the shape and size of dialog boxes, the menu names, the icons, even the placement, size and names of buttons. (Hooray for resources!) These kinds of little changes are

not programming miracles, they are simply touches that turn an OK Mac program into a great one. These changes are also the kinds of things that all you Mac fans reading this article could think of in your sleep. I have no doubt that they upped our rating in *MacUser* Magazine by a mouse or more. (Steve Bobker, the Editor-in-Chief there, has said "If *MacUser* gave a rating of six mice, *Smart Alarms* would get it.")

Something else that we had to do when we licensed the product from the programmers was create a professional looking package, and write a professional quality manual. This is where friends come in. Creating professional packaging could easily cost over ten grand just to have it designed. We simply found a starving artist friend (they're all over the Bay Area) and put her to work, giving her nearly free reign. She loved the responsibility, and we think she came up with an outstanding design. And all for under a grand. We also completely reworked the original Australian manual. (*MacUser* said our manual was "both complete and exceptionally clear".) In writing it, we tried to always keep in mind certain basic concepts: Be as simple and clear as we possibly can. Do everything in very small pieces. Assume the person is intelligent but currently knows nothing. We did the entire manual with *Word*, *MacPaint*, *PageMaker* and a rented *LaserWriter*. I was fortunate to have a talented friend, who had written manuals before, do the bulk of it.

During all this we were also working on implementing our marketing plans. Apple may be helpful with this if you've got contacts there, or you've had a track record, or you've obviously got the quarter million. We had to use pure sweat to even put our press list together. We went to the library, dug up directories listing hundreds and hundreds of computer publications and phoned them all up to see if they ever reviewed Macintosh software. (What's your guess on "Agricultural Computing"?) This got us a list of about 85 publications as well as contact people in all of them. We sent out as many press releases as we could. We thought up "newsworthy" events about ourselves or our product and would write up a page or two - basically trying to keep our name in front of the Editor's face in hopes of getting something printed. One of these mailings was a specific offer to receive a free review copy. (Only about three quarters would even take the product for free, but if we get reviews in 60 different publications I won't complain.)

A BMUG officer, Steve Costa, once said to me "Put out a great product, do everything right, and don't focus on making money. You'll make it anyway." He's right, of course. Mac people, by definition, don't necessarily flock with the crowds. They trust themselves to know a good thing when they see it. If Imagine Software has "a good thing," the word will spread. I'm honestly trying very hard to see to it that we keep this philosophy in mind at every turn, realizing that I'll benefit as much as anyone from this.

For example, Imagine has put out a public domain program, called *Smart Alarm Solo*. We paid an East Bay MUG member, who loves telecommunications, to upload the Solo to every major Mac BBS in the country. In this way we can get a useful version of our program into the hands of anyone who wants to see or use it. (Like the release *Smart Alarms*, you type in your own message and set a time when you want it to remind you. When that time comes, your message appears onscreen, no matter what program you're running. In the Solo, you can only set one reminder at a time, as opposed to the huge limit of 1600 at a time in the release version. If a reminder shows up when you're busy, both versions allow a single key stroke or mouse click to instantly send the reminder away - to automatically return 5 minutes later. The release also has many additional options here. In any case, I can objectively state that even *Smart Alarm Solo* is vastly superior to the Alarm Clock. And it is free, folks. Check it out.)



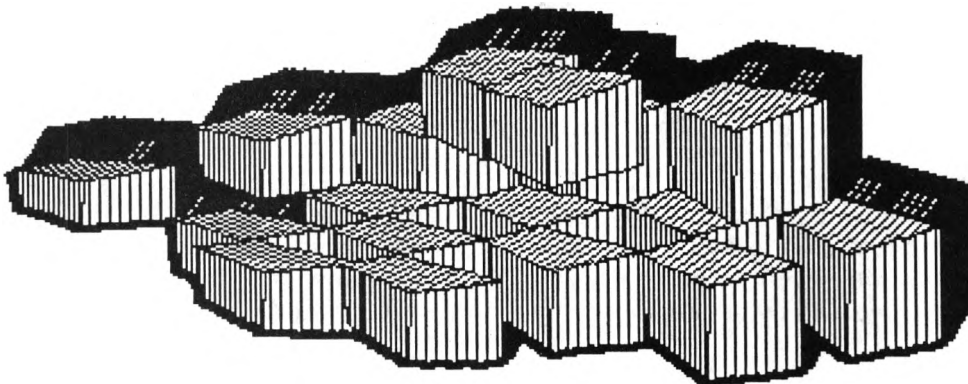
By the way, in line with 'doing the right thing', our product is not copy protected in any way. I personally feel it is the height of arrogance, not to mention stupidity, for a publisher to spend money on copy protection for productivity software. Not only does copy protection get in the user's way, but the hackers will crack it in a matter of weeks anyway. So why bother? Fortunately, many of the big software publishers have started to see this too. About time.

Of course, as a publisher I do have to keep finances in mind somewhere. Here, all that time spent in the User Group checking out new products paid off nicely once again. I had definite opinions about what our product should cost - both in comparison to other products, and what people would find fair. Our first financial decision was to lower the price of the package from the \$89.95 the programmers were asking to only \$49.95. At this price both businesses and home users could reasonably benefit from what we were offering. Of course, we also hope to sell more this way.

And finally, we do have to grease all the normal distribution channels. If we don't make an active push to sell our product through these channels, someone bigger or less conscientious will steal our ideas and sell them before we do. I believe advertising is a necessary evil in this regard. Dealers, distributors, mail order houses and even many businesses and individuals will not be interested unless the product gets strutted in front of them many, many times.

Imagine has done its share of advertising, but we've tried to be as careful as possible about stretching every dollar. If there's anything that eats up the budget, it's running ads. Well over half our entire budget has gone into some modest ads we ran last Spring and Summer. (By the time you read this, we'll be flirting with this fiscal vortex again by advertising our second product: *Multi-user Appointment Diary*. The new ads try to maximize our dollar by describing both the Single and Multi-user versions.) We've also printed up thousands of attractive and informative fliers, created mostly on the LaserWriter. People who request further information about our product, usually by circling the Reader Service Number under our ads, each get one of these. We're more than happy to send info to anyone who asks.

Fortunately, all this activity does add up to sales. It looks like Imagine Software is going to stick around, releasing what we hope to be high quality software products for the Mac. It didn't take a quarter of a million dollars. It did take lots of knowledge about the Mac, the ability to come up with useful or fun product ideas, a touch of professionalism, and a serious dollop of hard work and patience. And these are traits that many BMUG members might already possess.



A Review of the Excellent Exchange

Don Scellato

Almost every Macintosh user has heard of *Excel*, the most powerful spreadsheet program currently available. *Excel* combines spreadsheet, graphics, and database features. Everything that *Excel* does can be supported by macros. *Excel*'s macros may either perform user-designed functions or carry out user-defined instructions.

Programs that are as powerful as *Excel* are generally difficult to master unless the purchaser has had experience with other spreadsheet programs or database managers. When most people see the things that *Excel* can do, they are dazzled. When they attempt to put this power to work, they find that the learning curve is a steep one. A collection of *Excel* applications called **The Excellent Exchange** offers the new user an opportunity to learn *Excel* from tutorials. It also offers finished, useful applications to the more experienced user. Finally, as an exchange, it provides a marketplace for anyone who has written an *Excel* application that meets certain standards.

Guided Tour Worksheet					Guided Tour	
	A	B	C	D		
1		Projected Expenses			Welcome to the Guided Tour of Excel!	
2						
3		Jan	Feb	Mar	1	This program is designed for new Excel users
4	Rent	200.00	200.01	200.00	6	who want to become productive with Excel
5	Tel.	80.00	90.00	73.00	2	without spending long hours studying the
6	Heat	50.00	55.00	60.00	1	two Excel manuals before first starting.
7	Light	20.00	30.00	25.00		
8	Misc.	10.00	15.00	19.00		Its design is based on the premise that many
9		360.00	390.01	377.00	1	things can best be learned by doing them.
10						Format /Border
11		Database				
12		Apples		1		Select D12-E15. Select Format/Border/Top
13		Beans		2		Repeat the operation trying various combin-
14		Cake		3		
15		Dates		4		

Fig. 1. The Guided Tour shows you both instructions and a spreadsheet portion.

The Excellent Exchange currently consists of about 200 applications ranging from a Guided Tour of *Excel*'s functions, to a complete accounting package. Prices range from \$2 to \$90 per application, with most in the \$5 to \$15 range. Every application is supported by the more than 20 authors who have contributed to the **Exchange**. All applications have uniform quality and are internally documented. Many are supported by macro sheets that do most of the work.

Mailing Label Macro II							
Mailing Label II WS							
	F	G	H	I	J	K	
7	Mac Canada	34 E 29th St.	Toronto	ONT	M4H-1B7		
8	NY/NJ MUG	50 River Drive	Hartford	CT			
	Worksheet1			New Haven	CT	06520	
	A			Montclair	NJ	07043	
1	Mac Canada			Marlboro	NJ	07746	
2	34 E 29th St.			Cherry Hill	NJ	08002	
3	Toronto, ONT	M4H-1B7		70 Princeton	NJ	08544	
4				686 Edison	NJ	08817	
5				3205 Brunswick	NJ	08902	
6	NY/NJ MUG			Stat New York	NY	10128	
7	50 River Drive						
8	Hartford, CT						

Fig. 2. A mailing label macro.

Many of the applications make use of *Excel's* database features. One combination spreadsheet and database can be used to bill job time to a client; another can be used to invoice materials. There are tutorials that introduce *Excel's* spreadsheet functions, illustrate financial functions, illustrate how to build and use databases, and otherwise get the user started with *Excel*. One application builds calendars from scratch. Another sets up accounting worksheets. The list seems endless. If there is an application the reader would like to see, contact the **Excellent Exchange**. Ray Helzer, the man behind the **Exchange**, can write a custom application for you or can arrange to have one of the **Exchange's** authors write one for you.

Roots Worksheet																	
***** ROOTS OF EQUATIONS ***** This worksheet solves for ALL the ROOTS of the non-linear Quadratic, Cubic, & Quartic Equations that are commonly required in engineering practice.																	
Roots of 3rd Degree Polynomial (CUBIC Equation) $A(x^3) + B(x^2) + Cx + D = E$																	
• Where:	<table border="1"> <tr><td>A=</td><td>1</td></tr> <tr><td>B=</td><td>2.2</td></tr> <tr><td>C=</td><td>-2.95</td></tr> <tr><td>D=</td><td>0.7</td></tr> <tr><td>E=</td><td>-6</td></tr> </table>	A=	1	B=	2.2	C=	-2.95	D=	0.7	E=	-6	(If A=0, use QUADRATIC)					
A=	1																
B=	2.2																
C=	-2.95																
D=	0.7																
E=	-6																
• ROOTS:	<table border="1"> <tr><td>x(1)=</td><td>-3.5582417395</td><td>+</td><td>0</td><td>i</td></tr> <tr><td>x(2)=</td><td>0.67912086974</td><td>+</td><td>1.19237045166</td><td>i</td></tr> <tr><td>x(3)=</td><td>0.67912086974</td><td>+</td><td>-1.1923704517</td><td>i</td></tr> </table>		x(1)=	-3.5582417395	+	0	i	x(2)=	0.67912086974	+	1.19237045166	i	x(3)=	0.67912086974	+	-1.1923704517	i
x(1)=	-3.5582417395	+	0	i													
x(2)=	0.67912086974	+	1.19237045166	i													
x(3)=	0.67912086974	+	-1.1923704517	i													

Fig. 3. A worksheet that solves for roots of non-linear equations used by engineers.

If you've written an application for *Excel* and don't have a vehicle to market it, submit it to the **Excellent Exchange**. Ray will review it with you and possibly arrange to market it for you through the **Exchange**, and pay you a quarterly royalty on sales.

The **Excellent Exchange** is a step beyond shareware. It is a method of getting inexpensive, good quality *Excel* applications into the marketplace. Whether you are a person who wishes to learn *Excel*, a person who needs a specific *Excel* application or a person who has written an application you would like to share with others, consider the **Excellent Exchange**. A sample disk with the complete Catalog, a Tip Sheet and more than 15 Demos and Programs is available for \$4 from Heizer Software, 5120 Coral Court, Concord, California, 94521; Tel. 415-827-9013.

Msg. #4012 in **Public Messages** Posted on 10/24/86 at 13:48:16
 To: ALL From: DAVID ROWLAND
 Subject: Mac virtues
 In case you find yourself arguing with someone about the Mac, here is a good citation. Seafirst Bank in Seattle has bought thousands of Macs. Their micro manager was quoted as aying that the Mac is eight times easier to learn than the IBM PC. I suspect that is not an idle figure. With so much at stake I assume the "eight" means something, and I am impressed.



Microcomputer Consulting

Audrey D. Cole

Hi. I'm Audrey Cole, an independent microcomputer consultant in San Francisco. The editor asked me to write an article on what it's like to be a consultant. I worked for two major retailers for almost two years selling computers before I went independent just over a year ago. I work with Macs and PC-type machines. I set up small businesses and individuals with complete systems, train them, and hold their hands along the way. I don't sell any one piece of equipment for anyone, which keeps me truly independent. And I love what I do.

There are a lot of computer consultants out there these days. Some pretty good, and some who give this industry a rotten name. I can't really tell you how to go about becoming a consultant, because each one of us has a different story. I'll just tell you how I've done it and some of the pitfalls and tricks I've encountered along the way.

My initial decision was sort of forced on me, something that happens to a lot of people in the computer industry: the retailer I was working for went down the tubes. The whole decision took only about two weeks to make and went like this: I knew I was good when I started my business, otherwise I wouldn't have done it. That should be a given. If you're not really good at what you do, you have no business taking control of such an important part of somebody else's life. Your self confidence is also what will keep you going even through the inevitable hard times at the beginning.

Knowing people out there is the next most important factor. If you don't have any clients when you start, and don't really know any vendors or other reference-type people who can help you out, you're not in a very good position to develop a successful business (translation: you're gonna starve real quick!) And just for the record, having clients is the key to the "who you know" formula!

One place in which I'm very different from a lot of consultants is in being able to talk business—my client's business. Talking "techie" usually means instant death, because they're usually scared stiff in the first place. And you'd be surprised how little technical jargon you can use (we think we're talking normally, and we are—for us) before they start sweating and you lose them.

Use other specialists, both in fields related to yours to help fill in gaps in your expertise and in helping you with your own business. I use subcontractors who I check out very carefully for things like training on software packages I don't know. And I have a whole stable of good people I can call on when my work load gets too heavy. I also finally bit the bullet and hired a college student to do my data entry for me. Sure, I can do it, but I finally realized that I'm simply not going to, and that my time is better spent doing the things that I'm good at and that make me money. Don't be afraid to spend money on your business when it makes sense to.

Specialize. Stick with what you're good at and don't try to be all things to all people. Most consultants I know tend to want to try anything, either through desperation or because they think there's big bucks in it. Think hard about getting into markets you don't know about. (Do you really want to work on mainframes?) Think about how much time and effort you'll need to spend to get good at it, and if you're not better off doing more of what you do now. There will be a time when you'll want to expand, but do it carefully.



BMUG

FALL 1986

Don't forget about the business end of it! You can't just do your thing: you have to run your business, too. Big difference, and one that most aspiring consultants don't think about. (I didn't!) You must be good at sales, whether you like it or not, because unless you have clients ringing your bell night and day, what you as a business owner will be doing a lot of is selling. Selling your services and your products; selling yourself. And you have to be disciplined enough to keep good records (or rich enough to hire someone else to discipline you!) You should also have patience, and be good at teaching. Most of us seem somehow to spend a lot of time educating people, whether we planned to or not. It also helps to have some money available to tide you through your start up period.

I ended up being much more sedentary, and more isolated, two problems I had never thought about running into. And, though I hate to admit it, I get bored sometimes, and tend to wander. You really have to keep your energy and particularly your discipline on high when you don't have anyone cracking the whip over your head. Being your own boss is terrific, but requires lots of self-discipline.

If you're the kind of person who needs to know when the next paycheck is coming in, don't even consider being an independent consultant. Perhaps you can specialize for a management consulting firm or something.

Being a successful microcomputer consultant, a good one, is exactly what I want to be doing. It's scary, it's exhilarating, it can be good money. Most everyone I've talked with who has worked for himself (those who didn't starve, that is) would never want to go back to working for someone else. It's not for everyone, however. Put some thought into what working for yourself will be like. Do you have the stamina you need? Do you have a net if you fall? Talk to people doing what you are considering. The glamour is there, but so are the pits. Make sure you have a way out if you start floundering. After you've thought out the pros and cons, and you decide this is for you, go for it! And good luck.

Msg. #2115 in **Public Messages** Posted on 07/12/86 at 08:38:20
To: MICHAEL BEAUCAGE From: MICHAEL LAMOUREUX
Subject: Reply To 'AVATEX 1200 modem'
Here's the settings for an AVATEX 1200 modem (at least this is how mine is set). On the back, all switches are UP except switch 6, which is down. On the front, DATA/VOICE is pushed in, 300/1200 is pushed in (for 1200 Baud), and Off/On is pushed in (for ON!)>
Michael

Msg. #2116 in **Public Messages** Posted on 07/12/86 at 08:39:13
To: MICHAEL BEAUCAGE From: MICHAEL LAMOUREUX
Subject: Reply To 'AVATEX 1200 modem'
Opps, that was 300/1200 out for 1200 Baud.
Michael



BIG GUNS: *Double Helix* vs. *Omnis 3 Plus*

Rick Eisner

Four weeks ago, our project at UC was faced with the difficult decision to change from *OverVUE* to another database. *OverVUE*, which had been our first database, appeared fast at first look; but importing data from mainframe files became slower and slower as the size of the database increased. Ultimately, it was taking many hours to translate data into *OverVUE* files. Our principal database programmer accepted the assignment of determining, as quickly as possible, which database to use, and getting it working.

The office bought a copy of *Omnis 3 Plus*. (This demonstrates the feelings of some of our other programmers: they felt that *Omnis*, a character-oriented database like *dBase*, was workable, but that *Helix*, because of its icon-oriented language, would be too difficult for them to learn quickly). To encourage a balanced view, I arranged to get a copy of *Double Helix 1.0*, and our programmer agreed to exhaustively examine both over a weekend to determine which would best serve our needs. (We were hoping also to examine *dBase MAC*, which was announced to be released soon, but calls to the Ashton-Tate revealed that the release date would not be before the end of the year.)

On the following Monday afternoon, the decision was made: we will use *Omnis 3 Plus*. This decision was not made lightly, but, for our needs, *Omnis* had several advantages over *Double Helix*.

We needed to set up a turn-key system, which will allow unfamiliar users to easily access the database, while protecting all aspects of the program from their manipulation. *Omnis* allows this, including up to 9 levels of password protection (8 users and a master password). It is designed to allow end users to operate only the application itself, while supervisors can have access to limited aspects of the database. Although *Helix* allows custom menus to be easily set up and undesired ones eliminated, thereby protecting the software somewhat, it is still too easy for users to alter the program. And, *Helix* does not include more than one level of password protection.

Omnis is better self-documenting, e.g. it prints specifications of all formats, files, entry screens, etc. Data entry screens are easier to set up in *Omnis*, and are much easier to see. It is easy to show or print data entry screens do not clearly show you what column you're in, and it's hard to tell whether your data entry is within the space permitted in the form or if you've gone past the edge; the only way to be sure is to print the form. [A phone call to Odesta revealed that the latest version of *DoubleHelix* has database documentation—file formats, report forms, etc.]

Helix will not allow printing its graphic calculation screens on the LaserWriter (our only printer). This makes the calculations very difficult to document and to study. And, *Helix*'s calculation screens are very hard to read. *Omnis*' documentation prints out easily.

Omnis sorts very quickly on non-indexed fields—clearly faster than *Helix*. On indexed fields, technically *Omnis* is approximately 10 times as fast as *Helix*, though what we're talking about here is about five seconds for the slow one in a 2000 record file.

Speed of printing was a major consideration in our application. *Omnis* prints very quickly,



BMUG FALL 1986

about 6 pages of dense text a minute on the LaserWriter. *Helix* is much slower, partly because it allows complex formatting of pages. We have no final figures on its best time per page, but it appeared to take about half a minute per page.

In *Omnis* it is easy to print any files or reports you choose from user created menus. *Helix* can print only one report at a time, and requires that you open that report and pull down the **File** menu to **Print**. To print many reports would be very time consuming.

Only *Omnis* allowed setting up the program to run entirely from menus.

The size of our application compelled us to use *Omnis*. *Helix* keeps libraries which include format definitions with the data—in our application this took up an additional 125K with every data file—our application might keep six months of data (an additional 750K) until it could be verified and deleted. *Omnis* keeps data separately, and even then it takes up 790K.

Omnis can put numeric variables into character functions as if they were strings. *Helix* is fussy about what functions take what types, though it is easier to forgive this considering the number of functions it allows.

Omnis has command sequences. *Helix* appears to allow no macro functions at all.

Omnis allows many different layout entries, so parts of the same data can be seen on different screens. Judging from the documentation, *Helix* only allows a data item to be looked at in its original entry screen.

Omnis imports data very quickly. Large files downloaded from our mainframe, which were taking 7 hours to import into *OverVUE*, were imported in 20 minutes by *Omnis*. We did not have time to test this in *Helix*, however.

We got better help from *Helix* technical support at first. But we got better help from *Omnis* as we got more knowledgeable technicians on the phone.

On the other hand, *Helix* had some advantages, too.

Helix has a great selection of calculation tiles (i.e. functions)—over 120 of them. The wide variety and extensive abilities of calculation tiles makes it possible to set up almost any calculation within its language. *Omnis* has a much more limited set of commands: power, round, square root, and some primitive string, time and date functions—only 21 in all, and requires that they be operated within a click-a-box format.

Helix has great string and date and time functions, whereas *Omnis* is much more limited.

Helix can combine calculations easily and in intricate flow patterns. In *Omnis* the data entry sequence “language” is limited to a few executable statements (e.g. if, else, else if, repeat, until). The process of writing sequences becomes very tedious.

Helix had good documentation, although the improvements from *Helix* to *Double Helix* come as an additional manual. *Omnis* documentation depends on example procedures, and it is not always clear how they came up with their examples.



New *Double Helix 2.0* (the Multi-User compatible system) will allow some data storage space to be reclaimed by the system. *Omnis* won't allow files to be shrunk at all: it is necessary to export and re-import *Omnis* files to reduce their size, which can be a very time-consuming procedure.

The choice is obviously not completely clear. For a fast closed system, which my department needed, *Omnis* is superior. For an application which may change, *Helix* allows more possibilities for expansion.

Of course, having all these functions in one package would be wonderful. And *Helix* has promised another version soon, which may remove some of its limitations. This race is still in progress.



QGraphics/Cole

dBase Mac and dMac III

M. Joseph Schaller

There are two new full-featured, database products for the Mac worth taking a look at. One is a Macintosh menu version of *dBase III* from the PC world. The other is a 'new' dBase from the originators, Ashton-Tate, which takes full advantage of the Macintosh environment.

dMac III

dMac III looks almost identical to *dBase* on the MS/PC-DOS computers. Like *dBase III*, it uses command lines created from the 'dot' prompt, or stored in procedure files which can be run like programs. As in:

```
use mydata
index on coupon to mydata.ndx
list all price, yield, bondname for coupon > 10.000
```

All the interactive modes, such as **Browse**, **Edit** and **Append**, as well as formats for these commands, are supported. You simply do it all in a Macintosh window instead of a PC green screen. Instead of colons or inverse video highlighting the field names, they are surrounded by a thin black line on a white screen.

The person who has source code written in *dBase II* or *III* on any PC (and don't forget there exists more *dBase* code than anything else in the Universe except COBOL code) can now take that code, move it over to the Mac via wire or modem or network, and run the same programs on the Macintosh. All the features and commands appear to be here, with the exception of the MultiUser capabilities.

If you were hoping that Ashton-Tate would put out a *dBase* clone for the Mac, you should investigate this. The Ashton-Tate version is going to be a new *dBase*, and will involve substantial differences in implementation. Thus porting into *dBase Mac* may be slightly more tedious than using *dMac III*.

dMac III allows setting relations between files, allows use of multiple indexes, filters, format files, memo fields, program files, just like the real thing. File size is limited to only 32 Megabytes. *dMac III* uses its own index format and 'converts' *dBase II* and *III* files. It can read and write text, including *Microsoft Word* files.

I have only used a limited-demo version, which seems acceptable in speed and appears error-free. The Manual is modeled after the old *dBase* Manual, that is, "page 43 - **Create** command", in alphabetic order, and might be somewhat easier to digest than the real *dBase* Manual. The distributor says it has been on the market in Europe for some time and has been available in the U.S. beginning about June, 1986. It is priced in the same range as *dBase*.

Why would you use this rather than the 'real thing'? Well, first of all, it's available now, so if time is important... But the greatest strength is, if you have any kind of investment in *dBase* code applications and want to get something running on the Mac, *dMac III* is the obvious choice. Ashton-Tate's *dBase MAC* is a total rewrite, and although you can easily get it to read *dBase II* and *III* (as well as other) as well as data files, it does not use *dBase* program files. While a port will not be possible, there will be substantial time involved in translating and debugging.



However, that same strength means that using *dMac III* is in many respects equivalent to turning your Mac into a PC. *dBase* is notorious for being hard to learn, but powerful. That power is concentrated in data manipulation however, not ease of use or screen display. Since *dMac III* is a *dBase* clone, it shares these same characteristics to some extent, even on the Mac.

dBase Mac

The new *dBase Mac* is not just a porting of *dBase* from the PC, but a genuine (and successful) rewrite attempt to 'take advantage' of the Mac interface. This is the next generation of *dBase*. And they've done a good job.

dBase Mac starts off being different by being much more object-oriented. (Now, you think to yourself, I understand all that business about 'views' of the data in the new *dBase* on the PC. This is what they were thinking of.) Your data is essentially a collection of records seen thru **Views**, which are a collection of moveable fields on a screen that you can type on and move and cut and paste and even draw on.

What's different is that you now attach filters (show me only these records), before and after procedures, range-checking, and all kinds of characteristics directly to these screens or 'Views'. Yes, you can also write code, but since so much of what you used to do in code is taken up in these Views that, maybe 20% is left of your original 'procedure-oriented' code. You can still 'lock up' your code to produce custom environments for people without them fiddling with it.

dBase Mac is a relational database management system with hierarchical features. For instance, you can relate a list of names and donations to their corresponding phone numbers and addresses in another file. You link files together by drawing a line between them, much like *Interlace (Reflex)*, but *dBase* takes care of the pointers.

Links can be one-to-one thru many-to-many and one- or two-directional. There's a multiple-valued field, as in a Telephone Number, where you might have many values; they are all stored in essentially one scrollable field, which, if you wish can be made part of a linked set (as in address 1, phone 1, etc... or the line items in an invoice). In reports, the multiple values scroll out automatically.

There are many new *dBase* field types, among them graphic fields, into which you can put Draw or Paint images. You can control the mini-window borders. You can put graphics on your forms or reports. It looks easy to do menus, radio buttons, dialogue boxes and error-checking.

Something noteworthy to me: Scrollable buttons or 'pop-out' menus. One example would be a list of colors: The current color name is visible in the button; when you touch the button the whole list pops out above and below your mouse, even off the screen, and you move up or down the list. When you let go, it collapses back into the button, with your new choice visible. Putting all these things together is done in pretty much a MacDraw-like environment.

There is a rich set of built-in logic, math, string, error handling functions, again all on pop-out menus. More controls on the appearance of everything than you can imagine (this is a compliment), but arrayed well, not overwhelming. (You have to see these pop-out menus to realize how space-saving they are.) Non-programmers will be delighted to see how easy it is to put together a sentence in *dBase* now!



BMUG FALL 1986

dBase Mac can read data, so they say, from *dBase II, III, III Plus*, SYLK, DIF, text, ancient Babylonian wheat tallies, and so on. What's more, when reading from *dBase* files on the PC (think about a *TOPS* network), there's no conversion necessary. Even though it must keep its own files in a new format (due to its enhancements), it can still read and work with the old format from *dBase*.

How long should you wait? Ashton-Tate said that *dBase Mac* would be out before the end of 1986. I don't know how many bugs are left, but mine crashed in 10 minutes. When it is released, it should be a powerful product, although pricey.

What does it lack? Well, it's not... (what's the magic word this year?) MultiUser. And maybe won't be for some time. They said they are 'doing research' on multiuser capabilities. The generally accepted translation was: Let's see if Apple makes a standard. So if you need multiuser, Omnis (or Helix?) is still the contender.



International House, UCB Campus/Chan



Record Holder David Morgenstern

Have you noticed the many Mac users suffering from **BADS (Buying A Database Syndrome)**? You can see them at group meetings grinding their teeth at demos. Their copies of magazine reviews are covered with a million little notes and columns of cost differences. Can anyone (except recent Lottery winners) not empathize with these poor souls?

Buying any database is a heavy-duty decision. Look back at the ads in mags from a year ago. Remember *MacLion*? *LaserBase*? *1stBase*? *Habadex*? The price range of the 14 database products listed in the October '86 issue of *MacUser* extends from \$49.95 to \$495.00 with most around \$150 to \$200! The out-of-pocket expense is bad enough (I'm sorry Mr. Software Developer, but is any piece of software worth more than \$99.95 or even \$69.95?) but when you start adding up the time spent learning how to use the thing...! This summer my monthly cash flow (drought) led me to buy the cheapest database around, *Record Holder* by Software Discoveries, listing at \$49.95.

My angst was totally unjustified - *Record Holder* is a crackerjack database. It does all the sorting and computing you'd expect of any database with plenty of extras in text searches. It really follows the Mac Interface and is a snap to learn. Also nice is its lack of copy protection.

Here's a quick overview of what *Record Holder* can and can't do: It will do all the lists you want, and labels too! There's quite a bit of flexibility in the layout of reports so if you saved those extra-wide labels in the attic this may be of interest to you. *Record Holder* will not allow graphics in a field or in a report. A nice feature is the ability to preview the report on the screen. Some databases give you the choice of saving different report formats and data entry screens. *Record Holder* gives you only one data entry screen. Forget mailmerge and your LaserWriter. So what's so good?

Record Holder has an interesting way to quickly view your data in its "Table of Contents" window. Figure 1 shows a list of bird names in the Table of Contents window. You don't have to scroll through the whole list; just type a letter and you're at the first record beginning with that letter.

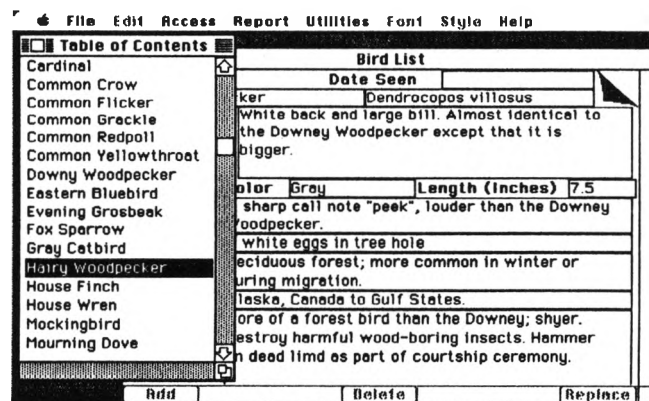


Figure 1.

Software Discoveries ads say this is the "Best database in the whole world". Give us a break!

BMUG FALL 1986

This is not a "do-it-all" database for Big Business, but it may be a great database for the typical Mac user. A tutorial included is based around baseball. The program comes with a set of templates for address lists, bibliography, expense accounts, sales invoices, checking account balancing, stock lists, and the bird list above. *Record Holder* text handling allows text fields up to 32K. Fields can be searched for imbedded text and allow the use of "logical operators" such as AND and OR. You can search any or all of your fields for the exact record required. The effort organizing and entering your data is meaningless if you don't have the ability to quickly and accurately retrieve your data. When working on a paper or thesis you can set up a database and input whole paragraphs of information, not just an abstract or the little arcane notes we used to put on 3 x 5 cards. Since *Record Holder* has plenty of room for text and lots of searching power, it is *better* to include those little things that don't really seem relevant at the time, but later are vital. When typing time comes it is simple to **Search**, **Copy**, and **Paste** from database to document using a *Switcher* set.

Here's a tip: After the input of a pile of notes from some source document, staple them and assign them a number. Input this number as part of the record in the database. Put the copies sequentially in a folder. If you need them they're ready, but they don't need to be in any fancy alphabetical or subject order.

Record Holder is very easy to learn and use. I helped a librarian start a catalog of a special rare children's book collection. She had never used the Mac before and *Record Holder* was her first application. In a half an hour she was off and abstracting. The hardest part was getting an understanding of click and drag! *Record Holder* has fantastic context-sensitive help screens. The program "sees" what you're trying to do and gives you the help. It saves you time having to run through the index for an answer. Figure 2 shows the help feature in action.

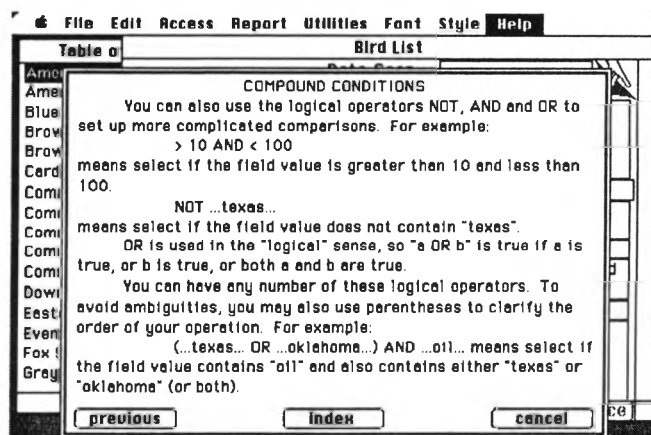


Figure 2.

Many of the Mac rags' reviews seem embarrassed about *Record Holder*. It's the old "You're a nice little database, but..." story. To get more respect Software Discoveries has raised the price from \$49.95 to \$69.95. It's still a good deal at the higher price and you can find copies in discount stores for \$35 - \$40.



Poor Tom: An Associational Database

Glenn Harris

Mac, do you got a lotta nerve! You nitpick about my misspelled words and sloppy sentence structure. But do I hear anything outta you when I'm beating my head, trying to knock out a good analogy for the ABM Treaty? You couldn't even give me an alliteration for the Secretary of State. You injection-molded Milquetoast!

And it's true for all the rest, too. Computers aren't creative—they aren't even imitative. But they might possibly be inspirational.

Some people claim that thinking can be enhanced through the practice of some very basic exercises. Special routines are supposed to help skirt "writer's block". One technique to jar that sticky brain involves something author Edward de Bono calls "random word stimulation". New ideas may be sparked from a randomly selected word picked, for example, from the dictionary.

Certainly, random words alone are not the key to creativity. The author must impose some goal and structure to this exercise or else it will quickly become little more than daydreaming. For instance, de Bono recommends choosing a single random word at a time and spending some fixed amount of time writing down inspirations from it. This notion of a balance between the random factor and structure provides a basis for the program *Poor Tom* which, hopefully, can be used as such an inspirational tool.

Poor Tom is a prototype for a "fuzzy set" database: one where records are related to one another by a measure both consistent and flexible. It does this by storing records as if each had a position in space; the relationship between records is thus based on distance. I also call it an "associational" database because the records are typically words, and their relationships are most easily understood by considering a person playing "word association." *Poor Tom* models that person's reactions by simply setting associated words *close* to each other, and dissociated words far apart. The program additionally allows you to change the distance between any two words.

Instead of working in just three dimensions, *Poor Tom* uses eight, one of which is reserved for word type (e.g., adjective). You can choose which dimensions to use in the various database activities. Thus, each dimension could relate records by different criteria: one could match words by meaning (a "Thesaurus" function) and another could relate them by their first letter; combining the two could aid in creating alliterations.

The relationships you set up are obviously distributive (A is near B, B is near C, thus A is near C), perhaps revealing provocative "found" associations. You could also use the program as an inspirational tool by looking for *dissociated* words—those uncommon combinations which would be the most likely to generate unusual ideas. And beyond just words, short phrases could be stored, producing more sentence structure-oriented inspiration.

Poor Tom is not an easy program to use. It's not useful as a tool until you generate a database, and the task of establishing word relationships requires patience and a certain amount of slack. You will never succeed in creating a file where all associations are absolutely logical. Nor is total disorder possible. The "space" of the program curves back on itself, so there are no absolute extremes in any dimension. No one record is more important than any other, each



BMUG

FALL 1986

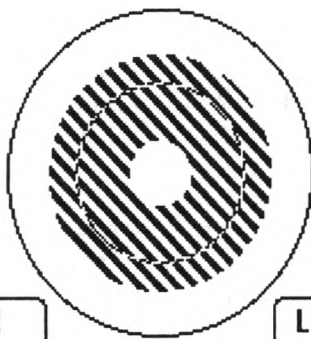
exists only to be measured against the others. The program imposes a type of moderation which almost amounts to a philosophy.

Poor Tom, along with instructions and a small sample database, is in the public domain and available on BMUG disk #44.

Searcher

Select

Fatalistic




Bind

Loosen

Include

Exclude

Search

☐ Judicial

☒ Fetching

☐ Forgotten

☐ Distant

☐ Noun

☐ Verb

☒ Adjective

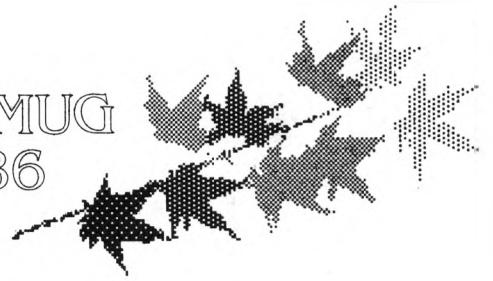
☐ Adverb

Msg. #2881 in **Public Messages** Posted on 08/29/86 at 12:45:54

To: ALL From: PHIL SUH

Subject: macs in tolman

UC Berkeley, just last week, got rid of all 40 Macs in tolman, and replaced them with 40 Mac Plusses all with 800K external drives. They have 2 laserwriters there as well, all hooked up via AppleTalk. If you're a student and need to use a mac or laserwriter, that's the best place to go...i believe the price is one or two dollars an hour with no charge for laser printouts...I'm not sure about this so don't hold me to it. Wouldn't it be great if we could all play mazewars in there?!!



Macintosh Statistics Software

Michael La Belle and Melissa Austin

Until relatively recently, one area in which software for the Macintosh was sadly deficient was that of statistical analysis. Now, however, a number of statistical software packages for the Mac have become available. For this newsletter, we originally intended to examine and compare six of these programs. Unfortunately, one statistical package (*Systat*) we wanted to look at because of its reputation as one of the heavyweights in the IBM world wasn't received by the time this review was due. Also, be aware that we did not have the time to check the numerical accuracy of the functions and statistical procedures in the programs we did review. Before buying or basing any conclusions on results from any number crunching program, you should spend some time working with it using sample data (from a statistics text for instance) for which you already have the answers. Hopefully our review will enable you to narrow your choices down to a specific program.

Most of the programs we tried have you enter the data as you would in a spreadsheet i.e. in rows and columns. After you have entered your data you can analyze it directly or transform (recode) your variables using built-in functions. The number of functions available ranged from 6 for *CLR ANOVA* to 40 for *NCSS (Number Cruncher Statistical System)*. All of the programs included at least one set of sample data to work with while familiarizing yourself with the program.

We found that these programs varied quite a bit in ease of use, power, cost and in the amount of statistical expertise that they assume the user possesses. In addition, some of the packages are meant for use in only a limited area of statistics. Some of the programs were written in Microsoft BASIC and in order to run they require either the Microsoft BASIC runtime system (all the BASIC programs supplied this system) or you can run them under Microsoft BASIC. Note that any program written in an interpreted language such as BASIC is going to be slow compared to one written in a compiled language. Also these BASIC programs, if run using the Microsoft BASIC runtime system supplied with them, will not work with *Switcher*.

One important feature of any software package is the manual. In the case of complex programs, which many of these statistical programs are, the quality of the programs manual is critical. Insufficient documentation of commands, lack of examples and/or unclear instructions can result in a flawed analysis, destroying the usefulness of even the most powerful program. Of the programs we reviewed, the manual for *NCSS* was the one we liked least. The manual, while extensive, isn't as clear as it should be and it is difficult to determine how to use a particular function.

A complaint we have for many of the manuals is that their indexes are simply not sufficient. Items or functions which should be listed either aren't or else unexpectedly appear as a subheading under some other function or item.

Although all the programs are capable of running on a Macintosh with two floppy drives, for any serious statistical work you will need a hard disk. The programs we review here were run from a 20-megabyte hard disk on either an Enhanced Mac or a Mac+, using System 3.2 and Finder 5.3.

CLR ANOVA

by Clear Lake Research, 5615 Morningside #127, Houston, TX 77005, (703) 523-7842,



Price: \$75.00 + \$3.00 S & H., Student version: \$23.95 at Kinko's courseware exchange.

License agreement: Limited to use on one computer at a time. Multi-user licensing available.

Warranty: Obnoxious. This is the typical we aren't responsible for anything "warranty" which seems to be currently popular among software manufacturers.

The program is not copy protected and technical support is available by phone, mail or on CompuServe. Two versions of the program were received for review. The first version is in Microsoft BASIC and requires the Microsoft runtime system, which is supplied with the program. The program will run under HFS, but you must keep all data files on the desktop and you must keep the BASIC runtime system and the ANOVA program in the same folder. In order to label the y-axis graph of data the program requires that a font (supplied on disk) called "SideWays" be installed in the System file. Data can be entered either by an editor which exists as a desk accessory or by reading in text files from other applications.

CLR ANOVA is not a general-purpose statistical program but is meant specifically for the analysis of variance (ANOVA). ANOVA is used to determine if several populations have the same mean for a particular continuous variable. The program comes with a 21 page instruction booklet with no index, but it's so short it doesn't really need one. You should know something about the analysis of variance and how to use it properly since the booklet will not teach you about ANOVA. The booklet is very clearly written and there is no trouble understanding how to use the program. The program can handle up to a ten factor ANOVA consisting of 5 within-subject and 5 between-subject variables and can handle samples with equal or unequal sizes. A particularly nice feature is the ability to represent interactions graphically. If your statistical needs are limited principally to the analysis of variance, then this is the program you want.

There were, however, several things we disliked about the program. First, the program will only operate on data which is in a file on disk, which means that you must enter data, save it to disk and then run an analysis on it. Although the desk accessory editor (a 12K accessory which is included in the disk system file and which looks like a version of MockWrite) allows you to do this faster than if you are using another application to make the file, it still makes the manner in which you enter and/or correct data unnecessarily convoluted. Second, if the program can't find the data file you specified, it quits and returns to the finder rather than returning to the program. Also, since the program runs under Microsofts BASIC runtime system the program absolutely requires the data file it is operating on to be on the desktop, the data file cannot be in a folder even if the folder is the system folder or the same folder as the program. Finally, as a consequence of being written in an interpreted BASIC, the program is slow.

A beta test copy of a new version of CLR ANOVA (ver. 1.1) was also sent. The new version is fully compiled and so doesn't require the runtime system and should also be faster. This was the program we first attempted to examine, however we were never able to get it to run. Whenever it was booted, the program would load, set up the menu bar, then give an insufficient memory warning (this on a 1-megabyte Mac+) and promptly return to the desktop. Presumably this problem will be corrected in the release version.

WormStat

by SmallBusiness Computers of New England, Inc., P.O. Box 397, 4 Limbo Lane, Amherst, N.H. 03031, (603) 673-0228

Price: \$79.00. Educational price: \$39.00.

License agreement: Limited to use on one computer at a time.



Warranty: Typical obnoxious nonwarranty.

WormStat (ver. 1.01) comes on a single disk. There are two versions of the program, one for a 512K Mac and the other for a 128K Mac. Neither version is copy protected and both will run under HFS.

The *WormStat* program instructions are short, coming in at 8 pages. The instructions are clear and to the point both for the 16 functions and for menu usage. Needless to say, the instructions will not teach you statistics, although they do contain short explanations of the use for the statistical functions. The functions in the program are the basic statistical functions, the type covered in the first few weeks of any serious statistical course. *WormStat* is the simplest and also the easiest to use of all the statistical programs reviewed. One reason for the ease of use is that, in a manner reminiscent of *MacPaint*, you chose the statistical function you want from a palette on the left side of the screen. The 128K version of the program can handle a maximum of 10 columns of 100 numbers while the 512K version can handle 10 columns of 1000 numbers. Data entry is either from a text file created by another application (such as *MacWrite*) or through a keyboard option available under the Data segment of the menu bar. Using another application to create the data file is definitely the best route since the keyboard option for entering data is slow and unwieldy and definitely needs work. It would best be used for correcting errors or changing a few values in a data file.

If your statistical needs are very simple and if you don't plan on using large datasets, this program is probably the one you want.

Number Cruncher Statistical System

by Number Cruncher Statistical Systems, Inc., c/o Dr. J.L. Hintze, 865 East 400 North, Kaysville, Utah 84037, (801) 546-0445

Price: \$79.00 + \$3.00 S & H.

License agreement: Limited to use on one computer at a time.

Warranty: Typical obnoxious nonwarranty.

The program disk is not copy protected and the program, like CLR ANOVA, is written in Microsoft BASIC. The program requires the Microsoft BASIC runtime system, which is supplied with on the disk.

The NCSS program comes with a 160+ page instruction manual. Despite the large manual you still need to have a through knowledge of statistics since the manual only gives a very brief introduction to each function. In addition, the manual is not clearly put together and it is difficult to determine how to use a particular function. Also, even though the index is 7 pages long it is still not complete enough and needs to be expanded.

We had mixed feelings about the program itself. It is potentially a powerful tool for statistical analysis. It has the largest number of data transformations available of any of the programs reviewed and possess some nice features such as interactive variable selection for regression analysis, which allows the user to add or delete a variable in a dataset being analyzed. However, we found the program to be difficult to use (in part because of the manual) and to not be user friendly. Also, because the program is written in interpreted BASIC it is slow and to run it under HFS you must keep all files on the desktop while the BASIC runtime system and the program must be in the same folder. Based on these findings we recommend that, unless you really need some of the functions and procedures offered and you also have a very thorough knowledge of statistics, you go with another statistical program.



The Data Desk

by Data Description, Inc., P.O. Box 4555, Ithaca, N.Y. 14852, (607) 257-1000

Price: \$175.00 + \$5.00 S & H.

License agreement: Limited to use on one computer at a time.

Warranty: Typical obnoxious nonwarranty.

The program is not copy protected and comes with a manual (over 180 pages) plus a 50 page booklet of supplementary notes. The notes contain information on update features plus a quickstart guide which can get you up and running on the program very quickly. The manual and the supplementary notes are very well done. The manual gives some, but not all, of the math used in calculating the various statistics. Of all the program manuals, the Data Desk manual comes closest to teaching statistics without the need for a supplementary statistics book. The manual contains a list of reference texts. The manual gives a thorough treatment of confidence intervals, including the Bonferroni adjustment for multiple comparisons. If you are a relative statistical novice, the manual alone makes this program your best choice. As with most of the statistical program manuals, the index needs to be improved.

The *Data Desk* program is as well done as the manual. The programs emphasis is on "exploratory data analysis" i.e. displaying and summarizing data. Because of this emphasis the program has almost no multivariate analysis capability except for multiple regression. The programs nice features are the ability to rescale a histogram, the zoom option on scatterplots, the boxplots, the calculation of skewness, kurtosis and percentiles, and the normal probability plot. The program also includes some unusual statistical procedures such as the ability to calculate biweight, generate random numbers and do simulations for a variety of statistical distributions. Unlike most of the other programs data can be entered either as text or numbers and actually entering the data is easy and straight forward. Data is entered in files consisting of one row by one column and various files can be bound into bundles for analysis, which is an unusual approach. In addition, the new version has a feature that, until now, could only be found in *MacSpin*. This new feature is the ability to graph your data as a rotatable 3-dimensional point cloud. Unlike *MacSpin*, the *Data Desk* allows more than one 3-D plot on the screen at a time and if there are points in the second plot which are also in the first it allows you to selectively highlight them.

Overall, the *Data Desk* is very useful for basic statistical analysis; however be aware it cannot be used for more sophisticated, multivariate analysis.

Recommended.

Statview 512+

by BrainPower Inc., 24009 Ventura Boulevard, Calabasas, CA 91302, (818) 884-6911

Price: \$295.00

License agreement: ?

Warranty: Reasonable. If you report a programming defect and it is verified by BrainPower they will correct it and return the program to you at no charge.

The program comes on two disks and is not copy protected. Technical support is available by phone. In addition to the *StatView* program itself the disks contain an unusual program called *Joiner*. *StatView* itself consists of two segments, which together occupy more than 500K of disk space, too large for the old 400K disk drives. If you have an 800K disk drive or a hard disk, the *Joiner* program will fuse these two segments into a single 532K application.



The *StatView* manual is 180 pages long and covers its topics clearly and concisely. You should have no trouble learning how to implement the various statistical capabilities. Unlike the *Data Desk*, the *StatView* manual definitely assumes you have a good working knowledge of statistics since explanations of when to use a particular function are minimal. The manual has a list of reference statistical textbooks. A nice feature is that the manual gives a thorough list of the formulas used in the statistical calculations in an appendix. The 180 page manual has only a 2 page index, which just isn't sufficient.

StatView 512+ itself is a good general purpose statistical program, although it is weighted towards factor analysis—a type of analysis which is used extensively in psychology. After NCSS it has the largest number of data transformation options and a very large number of statistical procedures. Unlike the *Data Desk*, *StatView 512+* allows the user to perform a considerable number of sophisticated multivariate statistical procedures, such as principal axis analyses with orthogonal and oblique transformations, factor scores, etc.

Of all the programs reviewed, the ease of use and sophistication of the data sheet in *StatView 512+* is simply the best. Alphanumeric data is entered into cells in a spreadsheet much as you would in a spreadsheet such as *Excel*. Entering data into a particular cell or choosing an entire column(s) or row(s) of cells for analysis is done by pointing and clicking with the mouse. Scroll bars at the edges of the spreadsheet allow the user to move quickly and easily around a large spreadsheet.

Recommended.

MacSpin

by D²Software, Inc., 3001 North Lamar Boulevard, Suite 110, Austin, Texas 78705, (512) 482-8933

Price: \$99.95

License agreement: Limited to use on one computer at a time.

Warranty: Typical obnoxious nonwarranty.

MacSpin comes on a single 400K disk and although it can be copied it does have an unusual form of copy protection. Every tenth time the program is booted, or if more than seven days have gone by since it was last used, the copy of the program will ask to see the master disk before the program will finish booting.

Strictly speaking *MacSpin* is not a statistical analysis program. *MacSpin* allows the user to make 3-dimensional plots of data and relationships can then be determined by rotating the data cloud and looking for patterns. *MacSpin's* 180+ page manual is very well done and tells the user everything they need to know to use the program. The manual's index was judged to be adequate.

The *MacSpin* program is extremely powerful and generally equals or exceeds the 3-D plot capabilities and features of the *Data Desk*. In addition, the 3-D plots in *MacSpin* look much more 3-dimensional than do the ones in the *Data Desk* due to a selectable depth cueing feature. However it lacks one capability that is present in the 3-D graphs in the *Data Desk* and that is the ability to put multiple graphs with shared data points on the screen. The data worksheet is set up along the lines of a spreadsheet, making entering or changing data easy.

Recommended.



BMUG

FALL 1986

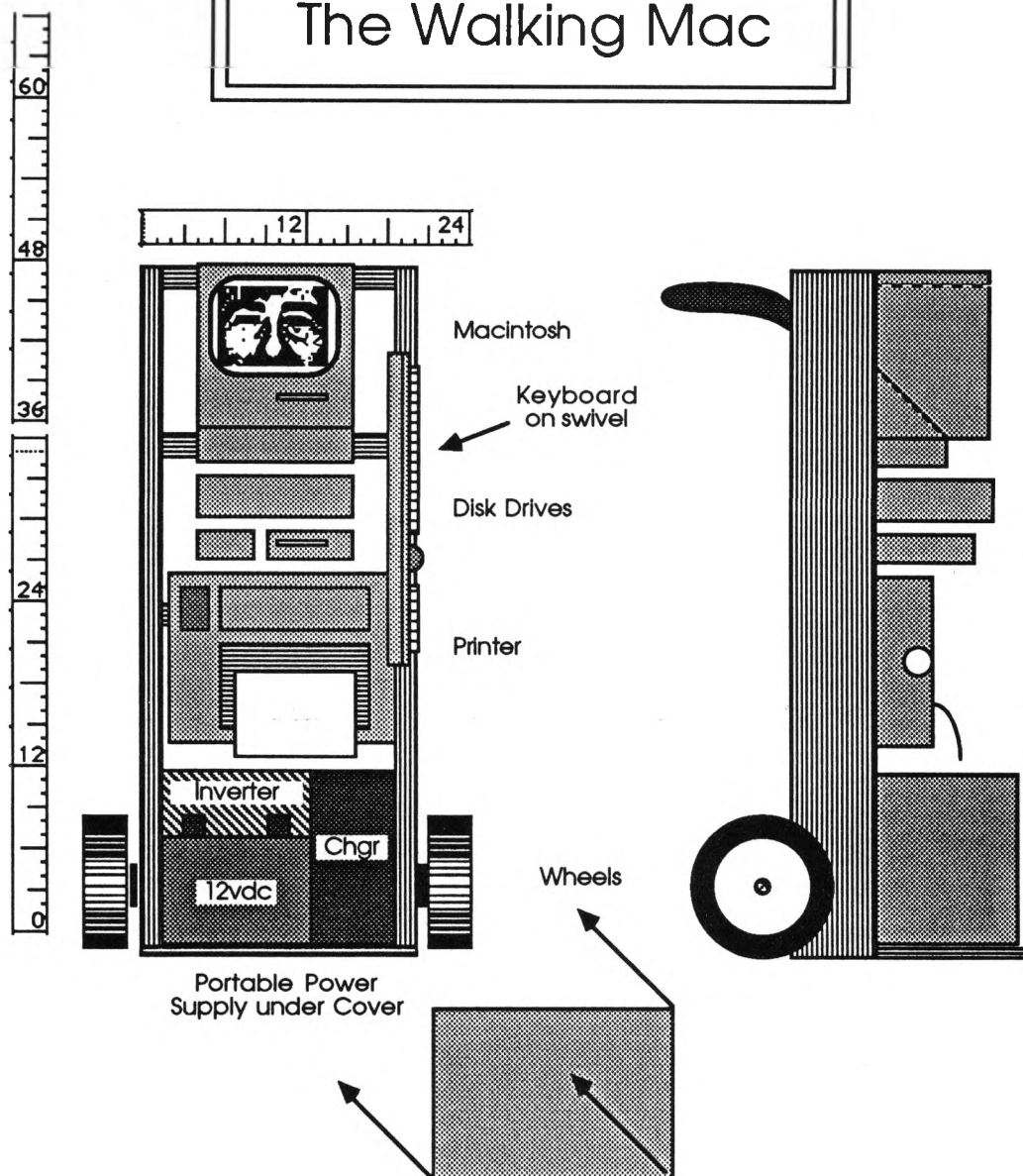
	StatView 512+	NCSS	WormStat	MacSpin	CLR ANOVA	Data Desk
Minimum Operating Requirements						
RAM	512K	512K	128K	128K	512K	512K
Disk	800K	800K	400K	400K	400K	400K
Data						
Maximum Number of Observations						
	22,500 (512K)		1,000 (128K)		-	
	65,000 (1,024 K)		10,000 (512K)		-	
		32,000 (disk based)		-		Disk based
Read/Save data as ASCII Files						
	Yes	Yes	No	Yes	yes	yes
Copy/Paste data from other applications						
	Yes	Yes	Yes	Yes	yes	yes
Search	No	No	No	Yes	no	yes
Sort	Yes	Yes	Yes	No	no	yes
Number of Data Transformations (i.e. Recoding)						
	35	40	10	31	6	39
Handles Missing Values	Yes	Yes	No	Yes	yes	yes
Outlier Trimming	No	No	No	No	no	no
Create Data Subsets for Analysis						
	Yes	No	No	Yes	no	yes
Univariate Statistics						
Mean	Yes	Yes	Yes	Yes	no	yes
Midrange	?	?	?	?	no	yes
Confidence Intervals	Yes	Yes	No	No	no	yes
Geometric Mean	Yes	No	No	No	no	no
Harmonic Mean	Yes	No	No	No	no	no
Median	No	Yes	Yes	Yes	no	yes
Standard Deviation	Yes	Yes	Yes	Yes	no	yes
w/ Variance	Yes	Yes	No	No	no	yes
Standard Error of the Mean	Yes	Yes	No	No	no	no
Coefficient of Variation	Yes	Yes	No	No	no	yes
Minimum & Maximum	Yes	Yes	Yes	Yes	no	yes
Range	Yes	Yes	Yes	Yes	no	yes
Percentiles	Yes	Yes	No	No	no	yes
Coefficient of Kurtosis	Yes	Yes	No	No	no	yes
Coefficient of Skewness	Yes	Yes	No	No	no	yes
Frequency Distribution	Yes	Yes	No	No	no	no
F Test of Variance	No	Yes	Yes	No	no	yes
T-Tests Paired Samples	Yes	Yes	Yes	No	no	yes
T-Tests Independent	Yes	Yes	Yes	No	no	yes
Graphs/Charts						
Pie	Yes	No	No	No	no	no
Bar	Yes	No	No	No	no	no
Line	Yes	No	Yes	No	yes	no



	StatView 512+	NCSS	WormStat	MacSpin	CLR ANOVA	Data Desk
Scatter	Yes	Yes	Yes	No	no	yes
w/ Error Bars	Yes	No	No	No	no	no
Box-and-Whisker	Yes	No	Yes	No	no	yes
Table	Yes	No	No	No	?	?
Histogram	No	Yes	Yes	No	no	yes
Stem-and-Leaf	No	No	No	No	no	no
3-D	No	No	No	Yes	no	yes
w/ Rotate Plot	No	No	No	Yes	no	yes
Color (ImageWriter//)	No	No	No	No	no	no
Contour	No	No	No	No	no	no
Normal Probability Plot	No	No	No	No	no	yes
Analysis of Variance						
Factorial/# of Factors	Yes/16	Yes/10	Yes/1	No	Yes/10	Yes/2
Unbalanced Designs	Yes	No	No	No	Yes	no
Pairwise or Multiple Comparisons (i.e. Contrasts)						
	Yes	Yes	No	No	yes	no
Repeated Measures	Yes	Yes	No	No	yes	no
Analysis of Covariance	No	No	No	No	no	no
NonParametric Tests						
Mann Whitney-U	Yes	Yes	Yes	No	no	no
Wilcoxon Signed Rank	Yes	Yes	Yes	No	no	no
Spearman Rank Corr Coefft	Yes	Yes	No	No	no	yes
Kendall Rank Corr Coefft	Yes	Yes	Yes	No	no	no
Kolmogorov-Smirnov	Yes	No	No	No	no	no
Wald-Wolfowitz Runs	Yes	Yes	No	No	no	no
Kruskal-Wallis	Yes	Yes	No	No	no	no
Friedman	Yes	Yes	No	No	no	no
Contingency Table Analysis						
Chi-Square test	Yes	Yes	Yes	No	no	yes
Degrees of Freedom	Yes	Yes	No	No	no	no
Total Chi-Square and Probability						
	Yes	Yes	No	No	no	no
G Statistic	Yes	No	No	No	no	no
Cramer's V or Phi	Yes	Yes	No	No	no	no
Tschuprow's T	No	Yes	No	No	no	no
Other Tests/Tables	Yes/Yes	Yes/No	No/No	No/No	No/No	No/No
Regression						
Pearson's r	No	Yes	Yes	No	no	no
Simple Linear	Yes	Yes	Yes	No	no	yes
Multiple	Yes	Yes	No	No	no	yes
Durbin-Watson Statistic	Yes	Yes	No	No	no	no
Polynomial	Yes	No	No	No	no	no
Stepwise	Yes	Yes	No	No	no	no
Residual Info	Yes	Yes	No	No	no	yes



The Walking Mac



This Macintosh should be easily transportable around the apartment, up to the roof, for springtime writing or computing out-of-doors, and would easily travel to a client's site for demonstrations. Of course, it weighs a little bit, but it's already mounted on a dolly.

Joseph Schaller

BMUG
Fall 1986



Commentary

COMMENTARY

Computers in Berkeley? • The Mac in local politics	Enrique Zambrano	361
Australia and the Macintosh • What's up down there	Ronald A. Schmidt	364
Piracy and Rentals • Legal update	Scott Beamer	366
Just Say No to Piracy • Nancy takes a stand	Scott Beamer	369
Diary of a MacNeophyte • How to be a free Beta test site	Glen Frendel	371
How Do We Pick Our Computers • Let me count the ways	Enrique Zambrano	379
A Beginner's Guide to Getting Help • Is there anything besides BMUG	Mark R. Chandler	382
Herb Cain Punches John Dvoruk for Making Pass at Dear Abbey • Ugly scene mars desktop publishing expo	Scott Kronick	383

Computers in Berkeley? Bits, Bytes, and the BCA

Enrique Zambrano

An Overheard Conversation between Jose RAM and Maria ROM at Juan's Place in Berkeley.

Two under-thirty student-types Huppies (Hispanic Yuppies) slumming at a back table. Intense and handsome, Jose is a recent Cal Ph.D. Graduate in Chemistry and Maria, a strong and sensual-looking woman, is an Anthropology major currently in her senior year at Cal.

Waving his hands and gesturing excitedly ,

Jose RAM - Maria, don't you see? I was dumbfounded when he agreed. There I was, mentally prepared to do battle with the City of Berkeley's chosen computer warrior, and he instantly destroyed my ability to soundly trounce his arguments by agreeing with me that **the Mac would be a far better computer for the City to buy for its various departments.** This from the man that had advised the almost totally ignorant representatives of the various City Departments that the IBM was a machine that would suit their needs. I had nothing left to say.

His explanation was that these people were so ignorant as to what was possible and available that it had taken 9 months of internal bickering just to arrive at the conclusion that a uniform operating standard should be adopted for the City. It had taken a full 12 months before that to get this group to agree that maybe word processors might be good for the more efficient running of the City.

Berkeley's well paid computer guru then informed me that maybe in **6 or 7 years they might be ready for Macs.** I mumbled that by then the Mac would probably no longer be the premier machine available. My parting volley was that for the supposedly "Progressive" City of Berkeley to adopt the IBM standard seemed completely out of character and that, as a reward for his brilliant advice to my City's Computer Selection Committee, I sincerely hoped that his hard disk would crash, without a backup, and that he could not find his copy of *FWB Hard Disk Utility*. I left his office, continuing down the hall, calling down various incantations of doom on his hardware.

Before I left the floor that his office was on I wished upon him eternal life in front of a CP/M-80 Machine with no manuals for the software.

Maria ROM - You're too idealistic, Jose. I love my Mac but that doesn't mean that everyone should buy one. I did read, however, that there are more registered personal computer owners in Berkeley than in all of the rest of the greater Bay Area

Jose RAM - You've got to understand that Berkeley is no Bakersfield or Orinda, even though sometimes I think that that is where some of our city employees come from. We're about to become the first city in the country to adopt, as a sister city, the "illegal" black township of Mathopestad in South Africa. Not only that but our own Berkeley boy Tom Blackaller is going to bring us the America's Cup from Perth. I personally think that he has plans to give it to the Cal Sailing Club at the Marina as a thank you for teaching him how to sail. Berkeley has a Worldwide Experimental and Progressive Image to maintain. We shouldn't be known as a City that is in the hands of a boring and obsolete IBM standard. **Besides which, I have come to the conclusion that it is politically incorrect to belong to the IBM camp.**



Maria ROM - You're beginning to sound totally rad Jose. What have you been smoking?

Jose RAM - I'm serious, Maria. I've lived in this town for over 9 years and I have a feel for when something is right or wrong. This town gets in your blood; you start to care even about things that don't affect you. If you look at the existing political parties you find that they are divided between Us and Them. **They are the Pin Stripe radical right extremists otherwise known as regular democrats or the ABC (All Berkeley Coalition). "Us" are the right-on, idealistic coalition of students, minorities, and act-locally/think-globally type environmental lefties known as the BCA (Berkeley Citizens Action).**

Maria ROM - What utterly divine madness! You remind me of the scientist in the new movie The Fly when he is talking about his experiments. Yes, Berkeley seem to provoke violent reactions from people. My father hates it and my mother loves it. Especially the ethnically varied restaurants. I think my father hates it because he is so straight that it seems like a foreign country to him. But what does political ideology have to do with computers?

Jose RAM - Don't you see the mental parallels between the operating systems for these two computers. The IBM MS-DOS system is the type of management style favored by the Corporate World and is full of rules and required memorization. This type of thinking results in compromise projects that look good on paper but are veiled corporate profiteering solutions that are usually bad for the long term health of the community. In the local case it resulted in the Golden Bear Development Project that was out of scale with the neighborhood and with the proposed Garbage Burn Plant which was sponsored by the ABC. Luckily we threw the rascals out before they could erect their burn plant and dismantle our local recycling crews.

Maria ROM - Actually I'm surprised that you almost seem to make sense in a totally far out context. OK, let's see what you can do to draw the parallel for the Progressive Coalition the BCA.

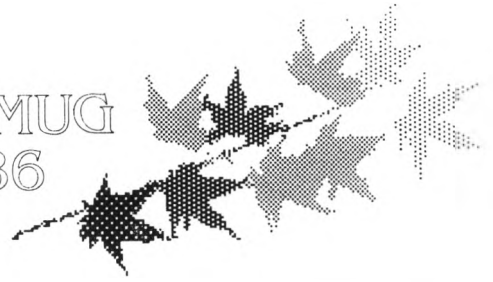
Jose RAM - Easy. Like the operating system for the Mac, the BCA style is intuitive, it encourages exploration. When mistakes are made, a large dialog box usually made up of local residents comes up before the council and makes a loud noise. This is often locally known as a public hearing. The council is then forced to deal with this issue before it can continue its intuitive exploration of what is possible to do while governing. District elections are a maximum dialogue box. After the students and BCA do some more compiling they may throw this measure out. The BCA theorists have shown a remarkable lack of sensitivity lately so they might misjudge how much the community has accepted District Elections. The BCA majority will then have to get some PhoneNet connectors, a hard disk, LaserWriter and a modem, all linked with Appletalk and Servant to communicate to the community all the great things it has already accomplished. And all the wonderful things that they wish to accomplish.

Maria ROM - Jose, at the risk of inflating your sexist male ego sometimes I have to admire the way your mind works. You almost seem to have an intuitive graphic Mac operating system burned into your PROMS instead of a normal brain.

Jose RAM - Thanks. What a great compliment. Let's go lowriding.

Running Citywide - Mayoral Candidate Loni Hancock has a Mac Plus
District 1 Incumbent Nancy Skinner has a Mac Plus
District 4 Incumbent Ann Chandler has a Mac Plus
District 5 Candidate Bobbi Singer has a Mac Plus and a LaserWriter

BMUG
Fall 1986



District 7 Incumbent Don Jelinek wants a Mac Plus
District 8 Incumbent John Denton has a Mac 512

*Enrique Zambrano - Entrepreneur - BMUGger - Tinkerer
Engineering and Building Contractor
Member, Berkeley Waterfront Commission
Chair, Restoration Committee to Save the Berkeley Pier
Berkeley Activist - Sailor*

Used MacDraft and Microsoft Word to prepare the Historical restoration specifications for the Cupola and Tower atop Berkeley's Original Town Hall. Recently purchased a second Macintosh for his two children, Alejandro age 8 and Christina age 5, they use MacDraft and FullPaint to draw Starrior and Insecticon robots.

DB 19 Connector Eureka? by Jay Capachi

A few days ago I took my 512 K Mac to my dental office and discovered I would like to move my hard disk a little farther away from my Mac. All I needed was an extension cord with 19 pin connectors at both ends. I almost gave up after calling 849-HELP, Apple, some local dealers and an out of state number Apple gave me.

Luckily a clerk at Al Lasher's in Berkeley mentioned a company with future plans to make DB 19 connectors. I called G C in Illinois at 815 968-9661 and found what sounds like the connectors I needed already being sold. One of their local dealers (Brill Electronics, 610 E 10th., Oakland Ca, 415 834-5888) has ordered the parts for me over the phone. I won't get them in hand before the newsletter goes to press, but they sound like what I want to make my own extension.

The female connector is part #43-125 for \$5.61 and the male is part #43-126 for \$5.84. The price could drop fast if at least four of a kind are ordered. Ken was the salesperson on the phone at Brill Electronics.



Australia and the Macintosh

by Ronald A. Schmidt

While spending 6 months in Australia working as a Macintosh support consultant for an Apple retailer, I had the opportunity to take a close look at the Mac community. It is a difficult task to summarize the state of Macintosh Down Under without over-simplifying, but I will present my observations anyway. All figures are quoted in US dollars.

Australia and Europe represent Apple's top overseas markets. As of Fall 1986, Apple has sold an estimated 20,000 Macs and 1000 LaserWriters in Australia (population 15.3 million).

Macintoshes exported to Australia are identical to US Macs except that they have a 240 volt power board. A 110 volt Mac will run on a transformer without a problem. I haven't seen a 240 volt surge protector -- they simply aren't used by Mac users Down Under.

Users Groups

The Mac Users Groups are tiny compared to their US counterparts. I've counted about 10 groups nationwide, with a combined membership of approximately 700. With the exception of the Apple User Group in Sydney, and the Melbourne User Group, most of the groups are still at the hacker stage, featuring occasional guest speakers and a BYOM policy (bring your own Mac). A few have bulletin boards, but none have regular walk-in office hours.

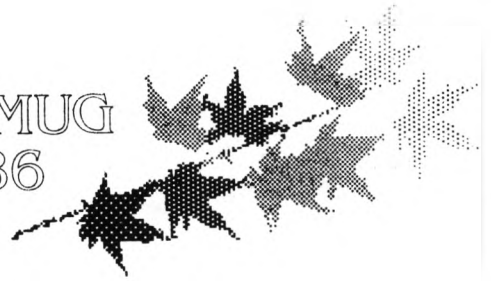
The most memorable Users Group meeting occurred in December 1985 when Steve Wozniak and Andy Hertzfeld spoke for two hours at the Apple User Group meeting in Sydney.

Prices and Availability

Local retail prices for Apple hardware tend to be about 25% higher than US prices. The price difference reflects tax and shipping costs, which add about \$400 to the price of a Mac Plus. Apple's new hardware releases usually arrive in limited quantities about a month after US release. However, some less popular hardware items, such as ImageWriter II AppleTalk cards, are consistently backordered.

The local retail price of Mac software and peripherals ranges from 20% to 200% more than US retail. In addition, many products aren't available from local distributors for weeks or months after their US release. Many factors influence prices that the local distributors charge, including: (1) Overseas air freight and courier costs, (2) Phone bills to US, (3) Customs charges, (4) Technical support and repair costs. Only Apple, Microsoft and a few Australian software developers have local offices and support hotlines for their products. All other software products are supported by the local distributor, due to the high cost and inconvenience of overseas phone calls.

The marketing strategy for the Mac has been to aim at the corporate user, i.e., engineers, architects, managers, etc. For the second consecutive year, Apple has featured the "Test Drive A Macintosh" campaign, allowing potential buyers to take one home overnight. The educational market is dominated by Apple II's, although the Mac has made its way into the universities. The Consortium, a purchasing option available through five universities, makes Macintoshes available at prices which are slightly above local dealer cost. Teachers and students can buy Macs tax-free if they meet the proper educational criteria.



Software Developers

Australian developers have completed roughly 40 commercial software packages to date. This includes five accounting packages tailor-made for the local monetary system. Other products include fonts, graphics packages, desk accessories, and templates. Several major product names are Fiscal (accounting), Smart Alarms (desk accessory), and Archipac (templates). All told, only a handful of public domain software has been exported from Australia.

The most successful Australian-made software package available in the U.S. to date is MacSurf, a three dimensional marine hull design program. Priced at \$2000, it is a package designed for the serious maritime designer.

Magazines and Expos

Imported US editions of *MacWorld*, *MacUser*, and *The Macintosh Buyer's Guide* sell for about \$7, arrive long after publication, and don't contain any local editorial content or advertisement. *Australian MacWorld*, published bi-monthly by the ComputerWorld Communications Group, sells for just \$2 per issue, and is very popular. Although affiliated with US *MacWorld*, and linked to the world-wide electronic ComputerWorld News Service, the bulk of material in *Australian MacWorld* is unique, and specifically relevant to Mac users in Australia. In other words, *Australian MacWorld* contains addresses and phone numbers within Australia, and lists prices in local currency! The circulation is approximately 12,000, which means that roughly 3/4 of the local Macintosh owners read it.

Australian MacWorld and Apple sponsored Australia's first Mac Expo, held in Sydney in November 1985. Although the Expo attracted over 5000 people, and was considered a success by both participants and exhibitors alike, it failed to return a profit for the organizers. At the present time, there are no plans for another Macintosh show in either Sydney or Melbourne.

Visitors to Australia who are interested in meeting local Mac users should purchase a copy of *Australian MacWorld* upon arrival, and call local retailers for the time and location of Users Group meetings. Be persistent. I'm certain that one of the 75 Australian BMUG members would like to meet you.

Australian MacWorld
ComputerWorld Pty. Ltd.
37-43 Alexander Street
Crows Nest, N.S.W 2065
Australia

Apple Computer Australia Pty. Ltd.
37 Waterloo Road
P.O. Box 31
North Ryde, N.S.W. 2113
Australia

MacSurf
Graphic Magic
P.O. Box 185
Cottesloe, W.A. 6011
Australia

Piracy and Rentals

Scott Beamer

Piracy

Piracy is like masturbation. No one likes to talk about it in relation to oneself, all say it's wrong, yet everyone does it. It's also like litterbugging. We are all responsible to keep the problem under control, yet the problem exists because of lapses by each of us.

Let's begin with the basics. What is piracy? How does it differ from copying other copyrighted materials? How is piracy different from copying with a Xerox machine, an audio recorder, a video tape machine? The answer... None. It's the same copyright law in each case. There are specific sections dealing with different media and precedent setting law cases, but the basics are the same. If you make a copy of material under copyright without permission, it is piracy. It doesn't matter whether you are copying recipes from a magazine for your grandmother, videotaping the football game to watch later, or making a tape of the record you just bought so you can listen to it in your car.

Look around you. Our houses, even our cities are hotbeds of piracy equipment and activity. See how your turntable and cassette recorder are connected together? What record are you planning on taping that doesn't have a copyright?

Why don't we use the word "piracy" to describe video taping, record copying, xeroxing, etc? It seems such a strong pejorative has become the jargon of the software industry, because of the frustration it feels for the profits illicit copies have prevented it from realizing. However, they must face it, piracy is built into the hardware, and copying ability is necessary for the machine to function. Software is a young industry, and very volatile. It's going to take time to sort itself out and blend its product more comfortably into the consumer market. Copy protection, rentals, and piracy remain on its rough edges. It must learn to live with the problem of illicit copies. All the older industries involving copyright have learned how to make a profit in this environment. The software industry does now and will continue to do so, as well. Records, books, videos all face this problem, yet we have public libraries, and essentially unrestricted copying for personal use in all these areas. I assume this age will come for software. Perhaps, it's already here.

This is not to say that illicit copies are not a concern of these other industries. Have you seen the paper that resists xeroxing yet? The record industry is lobbying to have congress require all imported cassette decks include a chip that will permit a sophisticated anti-dubbing system. The video industry has experimented with copy protection. But the primary focus of these industries is on foreign commercial piracy. Hopefully, as the software industry matures, their attention will turn to this large problem as well.

Shrink-wrap agreements on many products are very specific. Many claim that they are not even selling their product, but only granting a use license. Some state specifically that they are authorizing use of their product by only one user on one machine. Other shrink-wrap agreements are much more human and humane. Whatever the shrink-wrap agreement, the users go about their business doing what they think is right and convenient, largely unbothered by, and unaware of, what the shrink-wrap agreements say.

It is obvious that the software industry does not have in mind seeking legislation or enforcement of laws to prevent two students from going halves to buy a program or parents and



children from using a single copy of a program. Do they care if the child makes a copy and takes it with them when they go off to school? How about an architect's office with three architects and only one copy of *MacDraft*? Does the university have a site license for all the software in their micro computer room with forty computers? This is more the level of industry concern.

Of even more concern are local area networks and large companies where dozens and even hundreds of people have access to software. Though only one copy of a program may be necessary to give hundreds access to that program, in practice, more are required. Most large software houses make site licenses available in these circumstances. I suspect the business world, schools, and government are fairly straight about not abusing this situation. After all, it's not their money they're spending, and it could be very embarrassing for their image to be found out.

More sophisticated copy-protection is not the answer. Every time any of these industries gears up to try out a new system, the copy-protection breaking industry rubs their hands with glee and starts off on another round of cat and mouse. And even more dangerous to the manufacturer with a copy protected product is that his competition will not use copy protection and steal more of the market share.

There are many creative ways for the industry to deal with the piracy problem. Essentially, this is a question of modifying the product to fit the market. We seldom Xerox a book or magazine, because it's not worth the trouble to do so. We can buy the product at a reasonable price, or get it at a library. We don't dub records so much to save money. It's more for convenience. Besides, most people like the slightly superior sound of commercial recordings and the artwork on the covers. Libraries loan records, too. Do any public libraries loan software? I wouldn't be surprised.

The primary thing software manufacturers can do to combat piracy is improve the value of their product to the consumer. Better quality products are constantly coming out for the Macintosh, and will continue to do so. Prompt technical support should be provided, even "800" numbers by major manufacturers. Improving the quality of documentation certainly helps. Frequent significant upgrades should be offered free or at nominal cost. But above all, the most important is price. Prices are coming down and must continue to do so.

Much of the relatively high price of software comes from its evolution through the business market. Business is traditionally less sensitive to price than the consumer, because business passes the cost on to the consumer. It's the traditional capitalist way.

Rentals

One of my prime motivations in writing this article is to mention a bill recently introduced to the U.S. Congress. This bill, **The Computer Software Rental Act of 1986**, modeled on a bill passed two years ago, outlawing the rental of records without the copyright holder's consent, attempts to outlaw software rentals without the producer's consent.

From what I've read, the bill's sponsor, Rep. Patricia Schroeder has her facts wrong and does not seem very aware about computers or software. The good news is that it appears very unlikely that the bill will be acted on this year, but it is by no means a dead issue next year. I think it very appropriate that the congress hear from computer users, not just some producers (See **Mini Reviews** for a suggestion of how).



Software rental and record rental are not parallel services. One can hear music on the radio before buying it, or take it out of a library, or borrow it from a friend. Though software is even a hundred times as expensive, it is much more difficult to make an informed decision before buying. I have interviewed rental store operators, and they tell me a popular program almost never comes back, becoming a sale instead. It is the poor quality program, usually copy protected, that comes back again and again. I have reviewed two groups of a half-dozen programs recently. In each case, only the cheapest, poorest quality software program was copy protected.

Outlawing rentals will not change the copy-protection picture. In fact, one commercial programmer told me he was considering sending copy protected versions of his product to the rental houses. The chief counsel for the law firm that purportedly drove the first major software renter out of business with a lawsuit, saying his client wasn't against rentals in principle. They just wanted to assure heavy copy-protection was used and that the producers got royalties on the rentals. Both these people missed the nature of the rental business. One of its best features is that if the user is pleased with the product, they need only inform the renter that they are keeping the product. The convenience of this style of shopping is undeniable, and it is the current trend in retailing. Are not money back guarantees this same sort of arrangement?

Currently, rental outlets are rapidly expanding in number. Retail outlets are not dynamic. The typical store I have visited of late has had less than one hundred titles for sale, and the sales staff has practically no interest in demonstrating any but the most expensive program and to the high profile customer.

Rental by phone may become the major distribution outlet for software. Many walk-in stores (including Software for Less) are including rentals in their operations. Rentals are an interesting path between a market dominated by a few large, business oriented software houses and the alternative offered by Shareware.

Shareware was a failure as a commercial channel for almost all who released products in this way. It remains a channel of distribution by enthusiasts. If we look at a couple of those who have managed to do well financially in the Shareware environment, we see they used the same creative techniques we recommend to the major software houses.

There is a large enough market here for all these sources of software, including rentals, to co-exist. Users have a profound influence on the market. They can channel their influence to support those companies and individuals they prefer as software providers by being fair on the issue of piracy.



"Just Say No" to Piracy Scott Beamer

Nancy Reagan met with reporters yesterday in the Rose Garden, next to the White House. She was accompanied by two adorable young, white girls, Megan and Kelly. The First Lady had called this press conference to inaugurate her campaign, "Say No to Piracy."

It began at the regular meeting for "Wives of Important Republicans Bridge Club," Mrs. Reagan said. "While playing cards and eating cake, we women were idly talking about what we could do to help those people who really matter, those who vote Republican."

"Earlier that same afternoon, one of the ladies had been talking about how she knew her teenaged sons were software pirates, but she just couldn't catch them at it. She suspected her husband, who is a member of the House of Lords and one of the authors of the tough new legislation against piracy, had a secret software piracy problem of his own, as well."

"Well, it was just obvious to all of us what to do," the charming First Lady said. "We got Ronnie to promise us some money, and we had a pamphlet printed up on the signs of software piracy. Now we're all flying around the country, talking to concerned Republicans everywhere about this, trying to stop the epidemic before its too late. Why some of us are even taking time off from our busy schedules to go abroad and warn our good friends about the chances of this problem spreading to their countries as well. "

"Megan and Kelly are here with me today, because they are the first girls to turn their parents in to the police for being software pirates."

"Yes," said Megan. "At first, we didn't know our parents were pirates, but then we noticed that all our friends' parents' disks had commercial labels in pretty colors on them. None of our parents' disks had labels."

"Later, when we knew our parents were pirates," Kelly continued, "we couldn't ask our friends over to the house any more. There were xeroxed manuals lying all over the place. Our home life was terrible. There were late night phone calls from strangers. Our dad was gone most nights to 'user group meetings', he said."

Nancy's press secretary continued the narrative. "They girls knew they needed to get help when their parents started bringing home printed circuit boards and chips from flea markets. Then finally, they actually brought home an empty IBM case. They said they got it from a woman who had poured Drano into her husband's computer trying to get him to spend more time with her."

"The girls realized what was happening, because they had read of what to look for in the full color comic book they had received at the voluntary prayer meeting before school. This is the same one distributed by Nancy's bridge club. Their parents were into the hard stuff... They had become Hardware Pirates!"

"Her parents are now being held in one of the facilities used to hold Japanese American families in World War II. These are being refurbished to hold the first wave of detainees expected as the police step up enforcement of the Reagan Administration's new tough, no nonsense laws against piracy. More facilities will be needed later."

An IBM spokesman was also present to brief the press. "These girls parents didn't realize that what they were doing was undermining the American business way of life. All the layoffs at IBM are directly attributable to piracy. Luckily we have found a sympathetic ear in the president (though we had to present it in writing. His hearing isn't that good any more). He agreed to order congress to pass the legislation we wrote, and it appears it was done while there is still time. Much remains to be done, especially in educating the public. So few realize that every piece of pirated software leads us one step closer to being under foreign domination by someone like the insidious Japanese, who are already better educated than our Blacks and Hispanics."

"But let me tell you all, though it is true that software piracy almost invariably leads to hardware piracy, I have inspected the evidence taken from these girls' parents' home. You can rest assured. There was no chance that they could have ever have produced an IBM workalike. Their minds were probably so affected by their continuous use of pirated software, that they didn't realize the most important thing. To make a true IBM computer, you must include the IBM soul, and that's one thing no one is ever going to find in a flea market. Without the IBM soul, whatever the parents assembled could never be more than what it is... a collection of flea market trash."

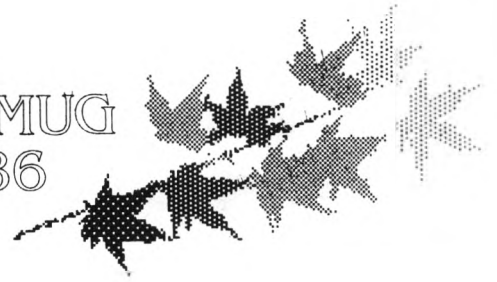
The press conference concluded with a photo taking session, with the First Lady surrounded by her aides, the little girls in front, all of them smiling and pointing at the camera as they sang, "Remember, pirate software... go to jail!"

Then Nancy joked with the press as she left, "Now you genlemen will have to excuse me. I have an appointment with my hairdresser's. You know, it's not easy being seventy five and still trying to look like a Barbie Doll."

While the press laughed appreciatively, she got in the presidential golf cart for the trip back across the lawn to the White House.



From one of the SunShine
Graphic Disks. ShareWare\$12
SunShine, PO Box 4351
Austin, TX, 78765



Diary of a MacNeophyte, or... How to be a Free Beta-test Site Without Even Trying © 1986 Glen Frendel

Dedicated to Walter & Marie

Foreword

I'm a professional musician who's been recording, and playing music for eighteen years. At present I'm the owner of a 24-track automated "state of the art" recording studio. I know about high tech and the attendant "angst" of high tech equipment. I thought after what I'd been through constructing my studio, and buying and interfacing the audio equipment I was ready for anything. From this dream-state my story unfolds.

The Dream

I purchased my Mac+ in early March of 1986. Two years earlier I'd considered the Mac and rejected it because of its limited memory, lack of expandability, and minimal software. Unfortunately, my insightful logic and clear thinking ended there. Instead of doing the practical thing and getting an IBM PC (or PC clone, which at that time was still a relatively rare breed) I ended up buying a 256k PC "compatible," A Tandy 2000 (which was a lot more machine than the IBM PC). I was right that an MS-DOS PC was the way to go. In retrospect the Tandy wasn't the best choice as a PC clone. Anyway that's what hardware fixation gets you. Mother always told me not to miss the bytes for the bits.

So on that fateful day in early March it was with great anticipation that I proudly unboxed my Mac+ and Imagewriter II. At last, I'd have a one meg machine (no more running out of rows in *MS Multi-Plan*, or incessant disk accesses in *Multi-Mate*). No more A> prompt. Aaargh! A machine with an interface that I could pick up quickly. A machine that would be a breeze to teach my colleagues to use. A machine that could do: word processing, spreadsheets, and graphics and actually print them (the graphics) out, unlike my PC. Maybe, dared I hope, I could use it as a database, to keep track of my financial records, inventory, mailing lists, etc. And the most important, indeed the most vital thing. Music applications. You see that's really what I wanted it for, what I'd been dreaming about. It would allow me to: record music, by entering MIDI events, enter musical notation much like word processing and edit (no eraser shreds all over the place, and legions of presharpened pencils on my drafting table), and transpose it (no more cursing the tenor sax player for not showing, and forcing me to transpose the part for my oboe playing neighbor, a snap with the transposition function on my musical notation software), and finally, print it out with near typeset quality.

Ah, the expectations I had that day in March. Now it seems a little strange as I think of that day and the hundreds of hours I've spent with my Mac+ since then. Hours that in the case of my dbase, which was created with an application which has proven to be completely unuseable, have vanished into thin air. It seems more like a decade than eight months ago that I enthusiastically plugged the Mac+ in and started the Guided Tour. Within two weeks I'd purchased: a SuperMac DataFrame 20 Megabyte hard-disk, Assimilation Turbo Touch trackball, Microsoft Word, MacDraft, MacCad, Versaterm, Helix, and Hard Disk Utility, as well as innumerable interface cables (many of which simply made it possible to connect pre-Mac+ hardware to the Mac+). Soon I'd also purchased ten BMUG public domain disks.



Experiences

BMUG

I walked up to the merchandising table. The next thing I knew a smiling Tom Santos was firing BMUG public-domain disks at me for five bucks apiece. He, like the group and its hierarchy, was friendly, helpful, and open. I felt like the spy that'd come in from the cold. A computer users group that felt like a revival meeting. It had an evangelical spirit. As if useful results in human terms, and creative enjoyment, were actually sought-after values which the group (congregation) thought were above crass commercial gain. Beyond the mundane world of the A> prompt. Particularly the Tandy A> prompt. A world that is (pardon me all you helpful souls who did your best to render assistance during my painful Odyssey through your Radio Shack version of the PC world) reminiscent of *Dawn of the Living Dead!* Computer Mormons?

The BMUG Disks

The day after I purchased them I tried to run them. I put them in the Mac, doubled-clicked on them and as I expectantly waited up came a dialog box saying "an application can't be found for this document." I thought, "Damn, I've been conned. That's why those disks were so cheap. That guy Tom sure was slick. He managed to unload these totally worthless disks on me for five bucks apiece. And they're not even double-sided double-density, so I can't erase them and use them with the 800k drive on my Mac+." No one told me I needed *MacWrite* (and it didn't come bundled with the Mac+, another switch by Apple) to use my BMUG disks.

Odesta

The trouble began immediately after purchasing the program (the problems with it and the System and Finder eventually forced me to reformat my hard disk). The dealer I'd purchased it from gave me the number of their local Bay Area Rep. She said (to paraphrase), "Oh are you using that version of the program! I wouldn't tell my worst enemy to work with that piece of dung, I haven't used that version of the program in a year!" "You should be using Version nn.xxx!" Besides, don't bother me, tell your troubles to the 'factory.'" (Thanks for telling me. Better late than never?). I mean, if it was no good, why were you (Odesta) selling it in the first place?

I received friendly support from the "factory," and was encouraged to purchase every version of their software, as a cure to my difficulties. Tech support was good. It was as if I was dealing with a schizophrenic. Sales would tell me in a very sunny way that what I wanted to do would be no problem, with Odesta's application. I'd buy the latest version of the program (each "improvement" an added expense to me of course, with the attendant transportation costs, at urgent rates because the more the program didn't work the farther I fell behind on my time schedule). I'd invest what became my standard number of hours, start to experience problems, call tech support, and be told "oh yeah, didn't sales tell you about that?" "Don't do anything serious with that System and Finder."

What about the reasons I bought this program for in the first place? It was a database that "doesn't require the user to have database programming skills." It was going to make it possible for me to generate forms, organize all of my miscellaneous data, and generally help get my infant business off the ground. I'd think; you're telling me "don't do anything serious with your program?!" Why didn't you say that in your glowing glossies? The ones that touted your program, raving about its incredible specs, claiming that it worked with HFS, and the new Mac+.



Real World

A small business man needs products that work, and doesn't have the time to debug or wade through the deficiencies of computer software. I didn't need the psychic pain of not knowing whether or not some mistake on my part was the reason the computer wasn't working, compounding an already painful and frustrating situation. Was I stupid? Unprofessional? Irresponsible? Illogical? Unmeticulous? No one I knew seemed to think so. I tried to be rational, careful, logical, follow the documentation, and do what the manufacturer's techs told me. And still the computer bombed. Did I have bad breath? Was it my mental vibes? Did it think I voted Republican, Democrat, Independent? They told me, "It's Apple's fault!" What's that supposed to mean to me? Try getting results from Apple, particularly when you're a neophyte Mac user, and you really don't understand what's happening. All I knew was I was in a bad situation. Odesta's unwillingness (and that of their counterparts, to the extent they existed) to make their product work, or refund my money, is the reason that the current generation of database publishers find such a leery consumer public. They're being forced to offer performance guarantees (money back if not satisfied, etc.), on products which in many cases are much better performers than their predecessors for a lot less money.

They are the victims of a deep and growing antipathy created by products that didn't work in spite of the claims of their manufacturers, and the manufacturers' unwillingness to "make things right." In my case, spending hours losing data files, and template layouts, only to find out "oh, that's because you're using a bad version of the System and Finder." I thought I'd gotten this machine so that operating it wouldn't require computer literacy. Unlike the dread blue box with the A> prompt. That's what all those ads said or implied. Each time I upgraded my database program, a new set of problems descended on me as if the purchase of my Mac+ was a commitment to join the order of the "Perpetual Sufferers Of Computer Purgatory." That fact was really maddening because I had to completely reconstruct the templates, in a not very friendly or easy to use graphics environment, one data field at a time (some templates had over a hundred data fields). My PC friends would say things like, "how many hours have you spent working on that? Why, that's trivial with *Dbase II*," "You mean you can't do that?" "It's a snap to write that procedure file on my PC," "It takes how long to search for a record?" or "You can't recover the data and repair the file structure after a system crash?"

The people at the Odesta said it wasn't their fault. That Apple had changed up on them, and hadn't followed their own System specs in designing the Mac+. There was nothing they could do. What the hell was I supposed to do? Why didn't they suggest (in those beautifully designed and carefully constructed glossy brochures) that purchasers of their product not attempt to create *anything* with their program until the "System and Finder" were finalized.

Denouement

After it became apparent to me that in spite of investing over three hundred hours of time going through crash and burn trauma over and over again, the application wasn't going to be useable. I called Odesta at their expense (as usual) over their 800 number. I reached a woman in tech support. She apparently (I was told later) was personally offended because I told her I was very upset with the way it appeared her companies product was working (or did I say not working). I was angry at having spent hundreds of hours of my, as well as my intrepid PC dbase expert Robert Baer's, time. A friend who's support if I'd purchased it, would have cost me hundreds maybe even thousands of dollars. Time watching my Mac bomb, and at the end of it all, having three forms that took three minutes to come up on the screen and if I entered any data into them waiting more minutes while they slowly went through their calculations. **Minutes!!!**

I'd done everything according to the manual (although my manual consisted of three different revisions and miscellaneous sheets) and consulted with their tech support I had no alternative), to be sure I was doing things correctly. My PC friends told me I was being very logical and approaching my database design exactly the way they would. This woman was insulted, although I hadn't personalized my feelings with her, (as a rule I wouldn't with any anonymous staff person), and explicitly told her I wasn't angry with her, but was simply fed up with having my time wasted, and wanted some recognition by Odesta of that fact. And, with that recognition, some sign of Odesta's willingness to make their product work. Obviously they'd decided they didn't like my "attitude" (an attitude which had developed after countless frustrating hours trying to use their product) because after the tech support person I usually dealt with called me back, mostly to tell me how I'd upset his colleague, Odesta hasn't communicated with me since. No update info, or anything else. Finally, on November 10, 1986, I received updated version "r35," dated August 26. Included was a warning about Systems and Finders previous to the current "stable" versions. The phrase "too little too late" comes to mind.

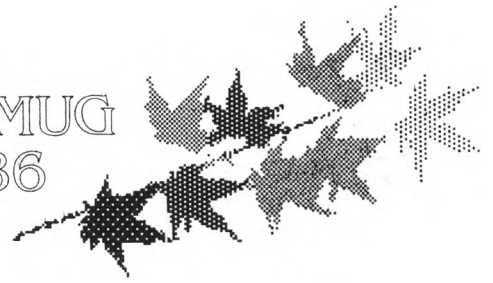
Database Madness

Ironically, my database madness peaked as Chernobyl was becoming a major news story. Somehow, at three o'clock in the morning as I listened to the news reports I thought I could hear the announcer saying: "Computer meltdown in Oakland." "Dbase bombs." "Mega-files eat recording studio owner." That turned out to be a horrendous bug in Apple's System and Finder (which is what Odesta maintained consistently while telling me, "don't blame us... get Apple to fix it." Yeah, right. Sure!). Of course Apple refused to acknowledge this bug through "official" channels, until after System 3.2-Finder 5.3 had been released.

SuperMac (Dan Clark)

In walks our hero, SuperMac's Dan Clark. He helped me many times (even on weekends and evenings). Working with me over the phone from the first day when I thought the DataFrame was broken because it was so noisy. They swapped it for a new one the next business day and later they told me there wasn't anything wrong with it other than the horrendous noise, something akin to a 747 taking off. On one occasion he gave me a System and Finder to try to cure my Mega-file problem with *Double Helix*. A software problem which SuperMac wasn't responsible for. He suggested he give them to me by phone. He spent almost two hours patiently resending the files while I tried to get my communications software configured properly.

I'm certain his efforts (along with the tech I was dealing with at Odesta) greatly accelerated fixing the mega-file bug. ("The Mega-File Bug." Sounds like the title to a Japanese Sci-Fi flick, doesn't it? Maybe they'll let me be an extra). Much of this help should have been unnecessary. Help I needed because of poor and or incomplete documentation, or buggy products, or for quite awhile because of Apple's own irresponsibility by making purchasers *defacto* Beta Test sites, for debugging the System and Finder. A phenomenon I encountered over and over again. That is, having difficulties because of incomplete, inaccurate, ambiguous, uninformative documentation, or simply undocumented bugs. This forced me to contact people at the software companies to find out what was supposed to happen. I've finally received what looks like a good trio of hard-disk software from SuperMac, for my DataFrame.



Software

Cortland

A print spooler that doesn't work, i.e. , crashes applications like *Excel*, *Double Helix*, etc. No warnings, and no notification after the point of sale. All I could glean about these bugs was what I could pry out of Cortland over the phone, which wasn't much in terms of effectively resolving the problem. They said they'd confer with Odesta to straighten the problem out (I actually instigated their initial communication). That was the last I ever heard of it. Their copy protection, combined with the self-installing nature of their desk accessories, may have been the reason my DataFrame became unuseable due to sector corruption. This was perhaps due to a conflict between SuperSpool (the DataFrame print spooler) and Blank, a screen blanking utility from Cortland. No one from either company warned me about any potential for conflict. Although Mac owners are supposed to assume that all DA's are suspect, somehow, experience as the best teacher is not my method of choice in this case. How this translates into practical terms, I'm not sure!!

Music Software

Opcode Software

Sequencer 2.0. A MIDI sequencer: The user interface is a difficult kluge at best. Their MIDI Hardware Interface has worked excellently in spite of the fact that Opcode had to do a hardware jury-rig to make it work without getting power from the Mac. Opcode said that was Apple's fault because there had been no warning about the elimination of power from the Mac's serial port. The *DX Patch Editor*, and *Patch Librarian* (both for the Yamaha DX-7) have both proven to be very useful programs. I found the owner to be highly intelligent (although inscrutably so), and imbued with the "Mac Spirit." His staff was helpful and efficient. Their recent bulletin was helpful in that it attempted to clarify the distinction between updates and upgrades.

Digi-Designs

I called to report a bug. At first the tech person I talked to said it was a problem with the hardware I was using. After explaining the various tests I'd performed to isolate the bug, he admitted (under duress) that there was a video refresh bug when using the Mac+. I also must say that the person I talked to spent a considerable time (mine as well as his. Of course by then I'd come to understand that the software companies thought at best my time was worth micro-dollars) making sure he understood the problem, and attempted to be helpful by suggesting a trick involving the use of the calculator desk accessory. By placing the calculator over the value fields that were supposed to contain values no greater than two digits, (which in reality were commonly getting scrambled up into strange four digit numbers) and then closing it, the video refresh would update those value fields by writing the correct values over the incorrect ones.

Digi-Designs never officially notified me of the problem. I wasted many hours of my time trying to work around what was wrong with the program. The tech told me that the latest version of the program fixed the problem on the Mac+. When I suggested they send me an update (which, in these cases, was **required** in order to fix a bug in their application, **not** simply to give me the benefit of more features), I was informed that company policy was that to get the update it would be necessary for me to send in **both** (the two) program disks. I asked what I was supposed to do without the application in my busy studio, where the program was being used heavily, and explained that sending both disks in would be a functional and economic hardship. I also suggested that having both versions of the program in my possession would make it possible for me to document the bugs I'd encountered and possibly assist in fixing them.

His response was "that's company policy." If that was company policy, why wasn't it clearly stated at the outset, **before** I purchased the program?

Mark Of The Unicorn

Mark of the Unicorn is the only company that has actually sent me a "free update," with an informative, useful and accurate piece of documentation. *Performer* (a MIDI Sequencer) is an excellent package and should be emulated by other Mac software makers for usefulness, ease of operation, use of the Mac interface, and cost/benefit ratio. It has paid for itself many times over.

Unfortunately, this isn't true of *Professional Composer* (a music notation program) which is slow, idiosyncratic, and, contrary to the claims of the company, won't print properly under HFS (at least, not running from a SCSI hard-disk. It is still one of the best programs of its kind running in any generic computer environment, and has the potential to be a first-class application.

Dealers

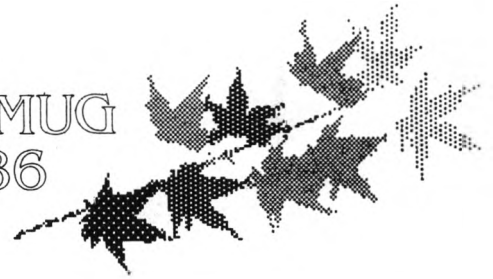
I quickly learned not to expect a dealer to have more than a superficial understanding of the way a program worked, although on more than one occasion I've been impressed with how knowledgeable and, generall speaking, willing to help some dealers are. This is definitely an improvement over the PC world. It seems to be the case that generally, Mac products dealers are more open, helpful, and free-thinking than their PC world counterparts. Possibly because they're genuinely interested in the Mac and how it works, as opposed to simply trying to squeeze every dollar out of the end user.

At other times, particularly when dealing with software companies, I felt that I was a victim of the business version of the "neoconservative" Reagan revolution. *Caveat emptor*, or roughly translated into commercial English, "up yours."

ComputerWare (source of my software & hard-disk)

They were generally helpful and friendly. Although I spent over fifteen hundred bucks, the salesman assured me that my purchase was trivial, compared to the kind of sales they were regularly making to Stanford. When I did my inventory, their bookkeeper was very cooperative in supplying a xerox of my original purchase receipt when I couldn't find my copy.

Sometime after selling me *MacCad Electronics*, Computerware decided that it was not a good value. After discovering they'd sold me the wrong version (they sold me the *MacDraw* version) for *MacDraft* and they weren't willing to expedite assisting me in getting a new manual and software (unless I wanted to wait three weeks). By calling the manufacturer directly (who was very responsive and helpful) I was able to remedy the problem in a week. If *MacCad Electrical* was a bad product, why did ComputerWare sell it to me in the first place? And once they sold it to me, weren't they responsible to service me in a reasonable and timely fashion? I.e. , by swapping for the correct program or arranging with the manufacturer to send it to me by UPS. Particularly since they sold me the (wrong) *MacDraw* version in the first place, even though they'd recommended (and on their recommendation I purchased) *MacDraft*. They were fully aware that I had a Mac+ and had advised me that it didn't come bundled with *MacWrite*, *MacDraw*, and *MacPaint* (which meant I had to buy those applications or comparable ones). They blew it... they should have been more cooperative in speedily resolving the problem.



Summary

(The Software/Hardware Manufacturer-Dealer-User Infrastructure)

Software makers should be held accountable for how they market and represent their products. The customer deserves good treatment. After all, customers are the force that drives the marketplace, and by making their purchases they make it possible for the software and hardware industry to exist. In spite of the generally friendly and supportive attitude of people involved in commercial endeavors with the Mac, I've never gotten the impression they weren't in it for the money.

Copy Protection

The necessity (at least according to certain manufacturers) for copy protection (an issue with many pros and cons) is a reaction to the tendency of Mac software users to trade and swap (and occasionally outright steal) software. This tendency which is largely an outgrowth of the users' fear (in many cases justified) of purchasing useless software, and not because Mac users are dishonest cheapskates. These products, in general, don't live up to the makers' claims, or fulfill the purchasers' expectations, created by the makers' sales hype.

For example, the purchaser should be (but rarely is) informed of all conditions (as could be reasonably foreseen by the maker) that pertain to after point-of-sale support by the maker. I.e. updates, revisions, bug fixes, etc. The difference between an "update," "enhancement," and a "completely new" version of the program should clearly be defined in the sales literature, along with the cost of any such support items, **before** the user purchases the product, not after. The potential purchaser should also be informed about any relevant considerations concerning technical support (whether or not it will be necessary), its availability, and at what cost, if any.

Future Mac

How will Mac users spell relief? "G-o-v-e-r-n-m-e-n-t R-e-g-u-l-a-t-i-o-n"? Is this the future of Mac software? Or software in general? Maybe so, if the producing side of the Mac equation maintains the current status quo.

The attitude that once a buyer has purchased a piece of software any problems he encounters are his tough luck, just isn't viable. Nor is simply installing an 800 number and offering free tech support. I'd be happy to pay for tech support **if** the application was well documented, so I could figure the program out for myself from the documentation, and the program actually did what it was advertised to do. The application must work and its limitations must be presented in a clearly defined manner. Not as "features" or "extras," as is the common practice. So the question as I see it is, will the manufacturers become self-regulated or will state and Federal agencies get involved in setting minimum standards of performance based on makers claims? Industry wide (i.e., including the PC world) . What can the user do to increase their clout re: independent software makers and Apple (which has been an incredibly arrogant, unresponsive giant to date). One question is: What will the resolution of the hardware competition between Apple and the independents mean to the user? Another interesting issue (that I think we'll be seeing a lot of in the near future) is the question of copyrighting a program's "look and feel." This will have far-reaching implications, especially in the Mac world.

Afterword

Promise

Now, after eight months, of on more than one occasion seriously considering (after an all night crash and burn session) entering a Buddhist monastery, and wondering whether there's



enough room on the planet for me and certain Mac software companies, the Mac is beginning to show its promise. The promise that drew me to it originally, and gave me the sense that here indeed was a machine that could do all the things that a computer should be able to do, without forcing the user to divest himself of his humanity, and adopt an approach that makes things easy for the computer.

I'm feeling more optimistic now. With a stable System and Finder, the software utilities that were supposed to be bundled with my hard-disk (when I purchased it eight months ago), the potential that a Mac with one megabyte of RAM (hopefully the RAM chips that make four megs possible will be available someday), an 800K floppy, SCSI interface, and the inimitable Mac user interface, the Mac offers users and software developers an environment that can support and utilize, useful, bug free, and sophisticated applications. The Mac+ is obviously a machine with a viable future. Hopefully a future a little more like the one I'd hoped for on that fateful day in March.

Good Mac Resources

Modern Accessories for Computers (M.A.C.), Berkeley: Ming Ho
A good dealer with good prices. Worthy of user support.

CJS Systems, Berkeley: John Sawyer

Informative and helpful. He went to great lengths (as a SuperMac dealer, but **not** the one who sold me my DataFrame) to give me the latest System and Finder, as well as the latest DataFrame software updates.

Software For Less, San Francisco

The owner is knowledgeable but the employees have a tendency of saying they have things in stock which in fact they don't (and couldn't have because they don't exist yet).

FWB Software, San Francisco

HD Util is a good product; the company is very supportive.

Music Software

A.I.C. Pro Audio, Concord: Matt Ward

Leo's Pro Audio, Oakland: Bill Thompson

Computers in Music, Daly City: Steve Nicholson and Joe

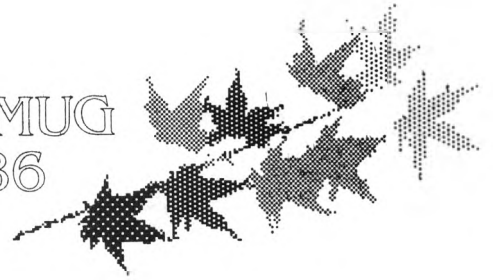
Computer Paramedics

BMUG groupies (Tony King, Tom Santos, Raines Cohen, Steve Costa, Steve Francine

Computers in Music: Robert Baer, Steve Nicholson

Odesta: Tim Fairbank

FWB Software: Norman Fong



How do we pick our Computers

Enrique Zambrano

How Do We Pick Our Computers?

The answer to this seemingly easy question can often be complex. For most of the estimated 90% of us who still have not joined in the technological revolution that has produced such marvels as \$4 digital quartz wrist watches (accurate to within 2 seconds per month), the answer is that we rely on our fears, prejudices, and hopes. Usually, since for our first venture into the computer world there is little logic to guide us, we have to rely on the opinion of a friend, a review that we casually read in a magazine, or often on a subconscious reliance on a brand name.

In short, most of us will buy our first computer without any knowledge of what else is available out in the real world. While the investment in a first digital watch can be taken as a somewhat casual affair (\$4), your first affair with a computer will, often as not, cost between \$1500 and \$2000. With this tidy sum you can usually have a computer with a second disk drive (useful when you need to make copies of disks) and a printer which is useful for when you wish to communicate to the world.

A Tutorial for the Computer Virgin

Looking through several of the computer magazines at your local bookstore can quickly overwhelm even a hardened computer crazy. The depth and breadth of new knowledge out there is overwhelming. There are even excellent free computer magazines such as *MicroTimes* and *Computer Currents*. Here is a simplified birds and the bees for the complete computer virgin.

First: The first computers were vacuum tube computers which cost millions of dollars and could only do simple addition and multiplication. Computer people often talk about "bugs" in their programs. This term originated in these early days when actual insects would sometimes cause an application to run incorrectly when the unfortunate creature would expire or worse in the circuitry.

Second : The space age produced the transistor which allowed us to replace the bulky vacuum tubes. Miniaturized components became available. This in turn made sophisticated circuitry more compact. The tinkerers and dreamers began to play in the labs and in their garages trying to approximate something akin to a home computer version of what technology had recently made possible in their laboratories.

Third : The first generation personal computers become available for the hobbyist market (under \$10,000). This generation of computer was not useful for anything other than perhaps blinking a sequence of lights. The dreamers see that with additional power and memory, home word processing is possible. We will never again be able to return to the simple typewriter world of inefficient word processing. Writing, editing, retyping, and producing a finished letter - taken collectively this is what is known as word processing when it is done on an efficient personal computer. I specifically mention word processing because 95% of personal computer use is for word processing (reports, letters, memos).

Fourth: The first computers become available for the lay person market and are controlled by 8-bit microprocessors. This means that they can send and receive information with 8 characteristics defined at one time. Defining 8-bits at a time enables the computer to define all

the letters A to Z and numbers 0 through 9. With a clumsy and hard to learn language, these computers prove amazingly popular because they put the power of the computer into the hands of thousands of lay people for the first time. The operating system or language for these 8-bit computers is known as CP/M-80. Looking at the title of the language you can imagine how much fun it is to have to memorize hundreds of commands like `PIP a:*. * b: /z.` Most of the Business community is not impressed and waits for further developments.

The next generation of home computers to arrive are the 16-bit microprocessors. Since they have more power and speed available the program writers strive with much effort to make programs that will appeal to the Business community. Graphics (charts etc.) and integrated software (programs where you can move from writing to relating columns of numbers and then on to other tasks) are now the new standard. The operating system is still clumsy and unfriendly. The operating system for this generation of 16-bit computers is called MS-DOS. It is still artificial and clumsy. I estimate that the average MS-DOS user has to study hard for 2 weeks to completely master each program. The very popular IBM PC is a 16-bit computer.

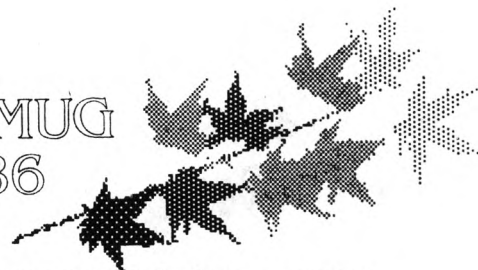
In order to avoid rapid obsolescence IBM leaves space for upgrade boards so that as home computers become more powerful the PC will be upgradeable and adaptable. Business and the corporate types, that I can only refer to as the **Suits**, go bananas. Here is a computer that will raise the productivity of the average secretary by 200%. An additional bonus is that since all the commands are artificial and alienating, i.e., `COPY a:*. * b: /z,` the computer is absolutely no fun so that no time will be wasted on frivolity. It even has an alienating screen display where what you type is shown as glowing letters on a black screen. The final and best reason of all of course is that if it's a respected name brand like IBM then all the **Other Suits** will congratulate you on your wise and conservative decision.

Fifth: The current generation of home computers comes on the scene. They are the truly amazing 32-bit microprocessor-controlled computers. With large internal memories, these machines favor the original vision of the dreamers and tinkerers that started the micro-computer revolution and that has now led to the user-friendly, interactive, personal computer. The Apple Macintosh computer is the leader in this market. The most amazing thing about this generation of computer is the friendly computer interface. Or, stated another way, the ease with which you can use the machine. Often when a new program arrives, the new owner will rip open the packaging and immediately begin to use the program (making a column or pie chart or playing a game of chess or drafting a document). What is remarkable here is that the user can immediately use the programs without needing to read any documentation. Perhaps the extraordinary accessibility of the software can best be illustrated by the look of wide eyed, mouth open, amazement on the face of a user of CP/M-80 or MS-DOS (who has just spent 2 weeks learning to use a program) when she sees you casually use software that is still warm from the package.

I recently had a female acquaintance tell me that she was ashamed to admit that she was a Computer Illiterate. She told me that she was going to buy a Macintosh this year and then set aside 3 months to learn how to use it. The poor woman still did not know that with the Macintosh she would only have to set aside three hours to learn how to use it. At least she had somehow become informed enough to want a Macintosh.

Summing Up

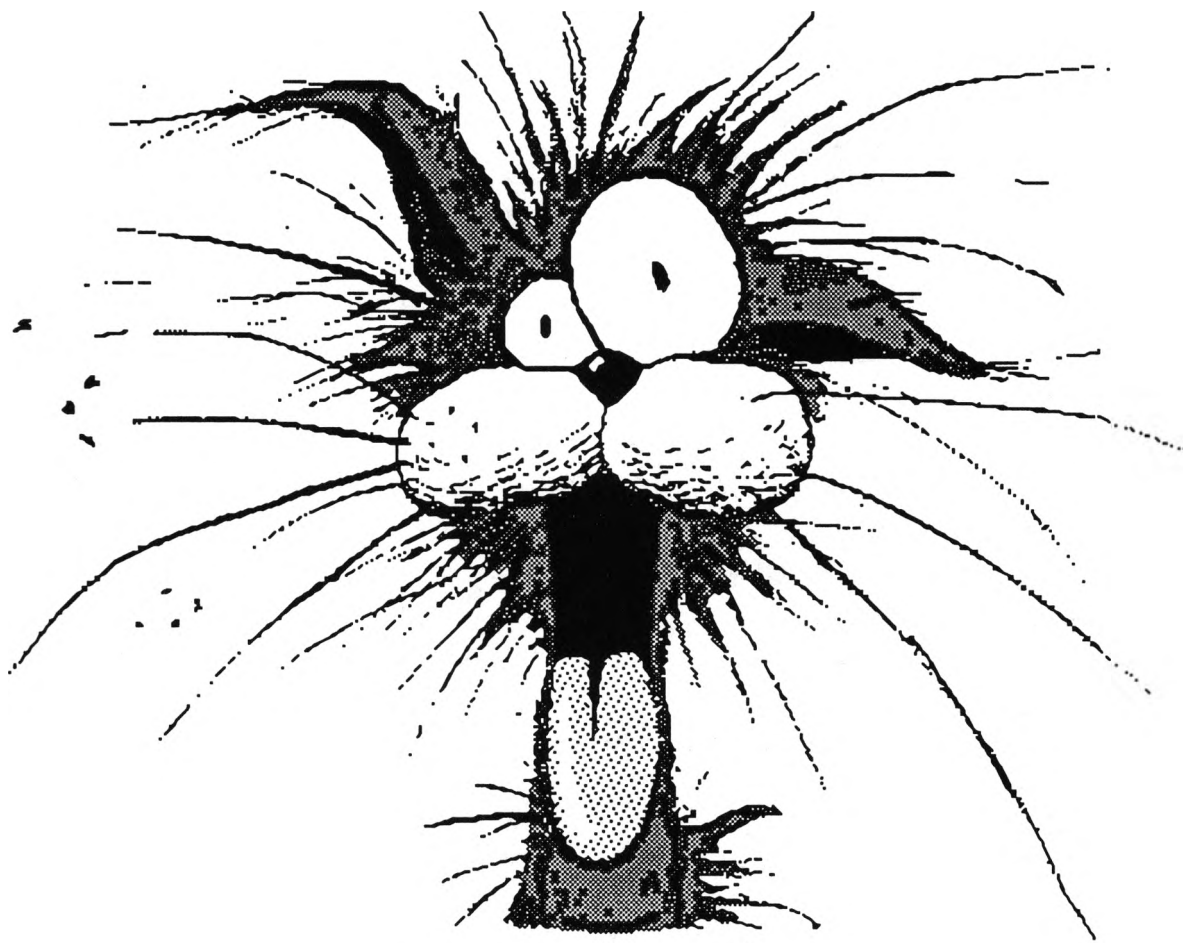
There is no conspiracy that is trying to sell us inferior computers. It is simply a function of the rapidly cycling technological times in which we live that operating systems and computers are often purchased by normal human beings with an almost total ignorance as to what is



technically available. I have had to teach myself not to gasp with amazement when a person who looks normal and apparently intelligent announces proudly that he has just purchased an IBM Clone for his personal use. Whenever this happens, I become depressed and am led to ponder the cosmic implications for mankind's survival, when otherwise intelligent humans continue to buy such outdated and barbaric tools.

When recently confronted with evidence of shoddy scholarship that went into the production of this article. Commissioner Zambrano's eyes narrowed and he mumbled what sounded like "May all your floppies be covered by magnetically charged pigeon droppings."

Commissioner Zambrano describes himself as a Californiano and a third generation Scoundrel. He insists that he be called Commissioner because he was recently appointed to Berkeley's Waterfront Commission and he is still very impressed with his new title.



Scanned and cleaned up by Phil Sue

A Beginner's Guide to Getting Help

Mark R. Chandler

When people ask me what I do for a living, I tell them I talk on the phone. Actually, what I really do is provide technical support for a local software publisher. That's right, I'm one of those people you talk to when you have a problem with your software. One thing I've learned is that if you want to get the most support for your phone call, you've got to be willing to do a little homework first. Below are some things I've found that make it easier to get good support.

One of the most important things you, or any computer user can do is to **register your software**. With some companies, in order to receive support, you must be registered. Plain and simple. It's also wise to save (or at least make a copy of) your receipt. Some software publishers offer a free update if you've purchased the product within a certain time frame, and they will usually want a copy of your receipt as proof. With Shareware, the entire transaction takes place in the registration of the program.

If you are having difficulties with a specific feature, or can't figure out how to perform a certain task, **check the manual**. Most manuals do a pretty decent job explaining how to use a program's various features, and most questions can be answered by looking there first. Some even go so far as to include a trouble-shooting section.

If you're still having trouble after consulting the manual, it's time to seek some outside help. Your best bets for assistance are the software company itself, your dealer, or your local users group (BMUG, no doubt). Remember earlier when I said you'd have to do a little homework, well now's the time. Before you make a call, or pay a visit, take a few moments to note the following:

- **Hardware:** Are you using a 128K, 512K, 512K Enhanced, or a Mac Plus? An external drive (400K or 800K)? A hard disk? What kind of printer? and so on.
- What **version** of the software are you using? If you can't find a version number (usually listed in the About... box, under the  menu), a creation date can be just as good (go back to the desktop and single click on the item's icon and select Get Info from the File menu, or use command-I). Also try to note the versions of your System, Finder, and printer files.
- Hard/Software **settings:** If you're printing, is it in Best, or Faster quality? Do you have your RAM cache turned on? If so, how much memory did you allocate?
- **Situation:** For example, "How do you draw lines in colors other than black?," or, "I get a system error when I paste a chart from my spreadsheet into my word processor."
- **Dialogs or messages:** If a system error does occur, what ID number is associated with it? If a strange dialog or message comes up, try to remember it, or write it down.

This may seem like a lot of trouble, but if someone does need any of the above information, you'll have it ready-leaving more time for getting your problems solved, and not scrambling for information. And besides, it will help to make you more aware of your computer and software.



Ugly Scene Mars Desktop Publishing Expo

Herb Cain Punches John Dvoruk For Making Pass at Dear Abbey Scott Kronick

Cain's Story

Real men don't process words. They bang keys on iron machines like the Royal upon this letter-strewn desk. If a real man makes a mistake, he uses real ink to slash it out. Why else would She have created union typesetters?

Last weekend, while searching for a public potty in a posh downtown hotel, I stumbled upon a non-event with the theme of Desktop Publishing. In mild discomfort, I viewed the latest in electronic gadgetry that I will not use. I distrust any machine whose manual weighs more than the salesman hawking it.

Listening to computerheads talking computerese is no way to spend an afternoon, especially after a liquid lunch. I've understood more asleep at a Portuguese performance of La Traviata. Worse yet, these people dress badly.

The affair was divided into two camps: The drab, lifeless clones from IBM (not the compatible computers, but the reps) and the scruffy street people with Apple (these guys still wear T-shirts). Couldn't one of these companies arrange a merger with Wilkes Bashford to exchange good taste for good accounting?

The straw that broke this cud-chewer's back was a Beserkeley-based user group booth that showed unparalleled distain for tailoring. The gentleman within me had no choice but to scoff, "Show me what makes your Macintosh machine so good that you can get away with dressing like slob."

Their head honcho looked at my name tag and smirked. "Log on, Mr. C." and he pushed the keyboard in front me. "Let our network make you a modern man."

To humor this oaf, I typed in my name and repeated the question. A few seconds later the replies started coming.

* * *

"I read your column all the time," writes Roy Thinnies, "except for the Sunday ones that would bore a cabbage to cole slaw."

Tracy Morgan: "The Mac has got font diversity, Herb. Even an old font like you ought to be able to see that."

William Pleth: "What makes the Mac different is its grace and geometry. Macs pour information like Chardonnay; other computers spill data like Bartles and Jaymes once through the kidneys."



"Is this the real Herb Cain, the gushy-tongued 'Frisco' columnist who's hopelessly stuck in the big band era?," asks Jon Roenicke. "Excuuuuse me, but last time I dressed formal, my black tie got jammed in the disk drive."

Nathan Shorett: "Glad to see you've joined the Mac world. I read your column every morning while on the can. Well, not every morning. I'd rather the delivery kid toss my Sunday Punch into a tub of heated Crisco than read your once-a-week attempts to wax poetic with SF nostalgia."

"Please pay no attention to all the nasty messages you have been reading on the bulletin board. We at BMUG are honored to have you log on. The enhancement of communication is our premier product and we think of you as a master of communication. P.S. Our membership fee is \$20 per half year and without it, we'll cut you off our BBS line faster than you can ellipsis your way to another subject." Thanks, Mindy Lorne.

"How can I get a job like yours?," Sylvia Steiner wants to know. "I'm great at opening mail and repeating other people's jokes."

* * *

Enough punishment, I made like the shepherd and got the flock out of there. But not before I witnessed a certain Brand X columnist harassing my favorite adviser beside a bank of bluish cathode tubes.

No. I never laid a hand on the man. Simply, I asked Ms. Van Burbon if she wished that I restrain the slug from his pedestrian advances. She replied that he was harmless so I left him unscathed. Why don't these Bloom County Banana 6000 creatures keep their silicon paws off the civilized gentry?

Dvoruk's Story

The hottest rumor out of Cupertino: mandatory drug testing for programmers. Apparently, the Pepsi baron wants to get back into the business of filling up bottles.

An industry insider who asked to remain nameless (his name rhymes with Gobs) says the idea for drug testing came from a committee of newly-hired middle managers who are fed up with the young, creative, talented types who get all the credit merely because they design the hardware and write the software. "After all, we're a marketing company now," the line goes.

Which brings up a number of interesting questions: Just how extensive is drug use among programmers? What types of drugs are being used? And the most fascinating question, one with kick-butt implications on the entire drug hysteria sweeping the nation: How the hell can someone perform in a profession that demands meticulous cognition while high as a Humboldt hummingbird?

The connection between creative expression and certain forms of intoxication has been debated as often as the Beatles' Sgt. Pepper album has been played. Here's a drug-by-drug examination from the programmer's perspective:

MARIJUANA — Insidiously addictive, yet remains the only substance whose use over the long-term does not inherently make you become stupid. Capable of spurring ingenious algorithmic thought, at least in those who prepossess a kernal of thought. Overwhelming the



programmer's drug of choice, though heavy use of California sinsemilla will weld your ears to your Walkman. Careful, the addiction is no joke.

COCAINE/CRACK — The 70's yuppie drug now has become the prime turf of scumbags, low-life, idiots, and other proud products of American cultural hegemony. The high's overrated. Regular use makes you greedy, witless, strung-out, and occasionally, dead. Programmers avoid all but a curious dabbling, preferring psychoactives for art and entertainment, and coffee for stamina.

PSYCHEDELICS — The big bopper, LSD, refuses to knuckle under after 20 years in the collective synapses. When the product is synthesized correctly, you still see God. Programmers can't resist that kind of temptation once a year or so. Tell a programmer it causes brain dysfunction and you'll hear, "You're telling a programmer about brain dysfunction." Nonaddictive because it scares the RAM out of you. Naturals go for the mushroom psychedelics and rarely mention disappointment. As always, it's travel at your own risk because you might be the next dude holding a cardboard cup and singing off-key Tony Bennett songs along Telegraph Avenue.

ALCOHOL — There's a decent chance some greasy, worthless, 9-to-5, lifelong loser is going to cross a divider and take away someone you love with splinters of metal and glass. Thank alcohol. Thank the drug-testing greasy, worthless, 9-to-5, lifelong losers who distrust highs in which you don't become stupid, loud, obnoxious, sloppy, smelly, and ugly. But you can't blame the drug because, in moderation, the ferment makes you and your mate tingle. Programmers use it well; many others use it to define a disease.

CIGARETTES — The nastiest addiction, bar none. The death is gruesome. People in the medical profession say the throat cancers stink so bad, they feel like puking while changing the bandages. The U.S. government subsidizes tobacco farmers. Nervous high school girls smoke lots. Drinkers too. Not many programmers. The intoxication is interesting; subtler than heroin.

DESIGNER DRUGS — Something potent could engrave your grey matter such that you'll be living a weird high long after the drug has been metabolized. Programmers can be brave, but usually will let Mikey try it first. Besides, chances are slight that your neighborhood chemist can compete with nature's finer herbs and fungi. MDMA is closer to dorky speed than ecstasy. PCP is like death with none of the benefits.

HAÄGEN DAZS, O'GRADYS CHIPS, ROLO CANDY — Substances deserving of BMUG's Choice Products List. Don't snicker at sugar, salt, and chocolate when talking about drugs. They pack a punch; sometimes a paunch, too.

What's a recreational drug? How's this: A peculiar food substance noted for its affect on mood and its inordinate wear and tear on internal organs. A seamless definition, and arguably appropriate as a Surgeon General's warning for popular fast foods.

Speaking of trashy taste, I was accosted at the Desktop Publishing Expo by none other than the Comical's famed columnist, who apparently suffered a fit of jealousy over my bantering with one Dear Abbey. With utmost respect for the elder statesman of hearsay and societal gossip, I recommend he stick to catered gatherings where they serve little hot dogs wrapped in greased dough.

Ironically, at this very exposition I witnessed an awesome instance of substance depravity. And, no, these weren't the people who make Macs; this was a much, much bigger company



BMUG FALL 1986

(whom I Better not Mention). A sad sight: a pack of darty-eyed, pin-stripers huddled over the competitor's LaserWriter inhaling from a toner cartridge.

Abbey's Story

Dear Abbey,

I never thought I'd be writing to you, but here goes. I am a respectable young man in my community. I work at a good job, go to church, perform volunteer work at the local hospital, and have a wonderful girlfriend who I plan to marry. My problem is: I had an affair with a Macintosh computer.

Now I feel awful. At first I thought we would just be friends. It was so cute and warm and friendly. Then one night we were up late together, fooling around. One thing led to another, and before I knew what happened we were telling each other sweet nothings, and well, you can guess the rest.

I don't know where to turn. How can I tell my girlfriend? What will she think of me? I can hardly look her in the eyes. The Macintosh sits in my den awaiting me. I know it wants more of my attention, but I'm a torn man. What shall I do?

I live in a small town where everybody knows everybody so please don't use my name. Just sign me...

Scott Kronick from Berkeley, California

Dear Scott,

I once had the same problem with my typewriter. Which isn't to say that what you did is right. You should be ashamed of yourself. You're a weird bozo and you need help. There's a group for people like you, but I understand Hewlett-Packard isn't hiring currently. If your girlfriend hasn't left you already, I suggest you beg her forgiveness and buy her a small foreign-built automobile.

Dear Abbey,

I'm an attractive single woman who can't seem to meet a nice man. I've tried everything. From church socials (everyone was married with two kids) to adult school (only two guys in the class and they were together) to smoky fern bars (the men smell bad). I'm not interested in casual encounters, and the idea of one night stands, while always repugnant, has become downright deadly. Where can I find a decent man to love and cherish to the altar and beyond?

Missing Someone Special in Oakland

Dear Missing Someone Special,

Men are to be found where men like to be. I know one such place near where you live. It's a club that meets once a week on the U.C. Berkeley campus. It's open to the public and costs nothing to attend. The attendees are exceedingly bright, friendly, and courteous. And best of all, the ratio of men to women has historically been 12 to 1.

It's called the Berkeley Macintosh Users Group (BMUG for short). Even if you think computers are the pits, feigning a small amount of interest ought to provide you with conversation, contacts, and an unobtrusive way to

BMUG
Fall 1986



meet guys who are above average in the brights department. They aren't all computer nerds; they're people from various fields with a common interest in learning about the most artistic of personal computers, the Macintosh.

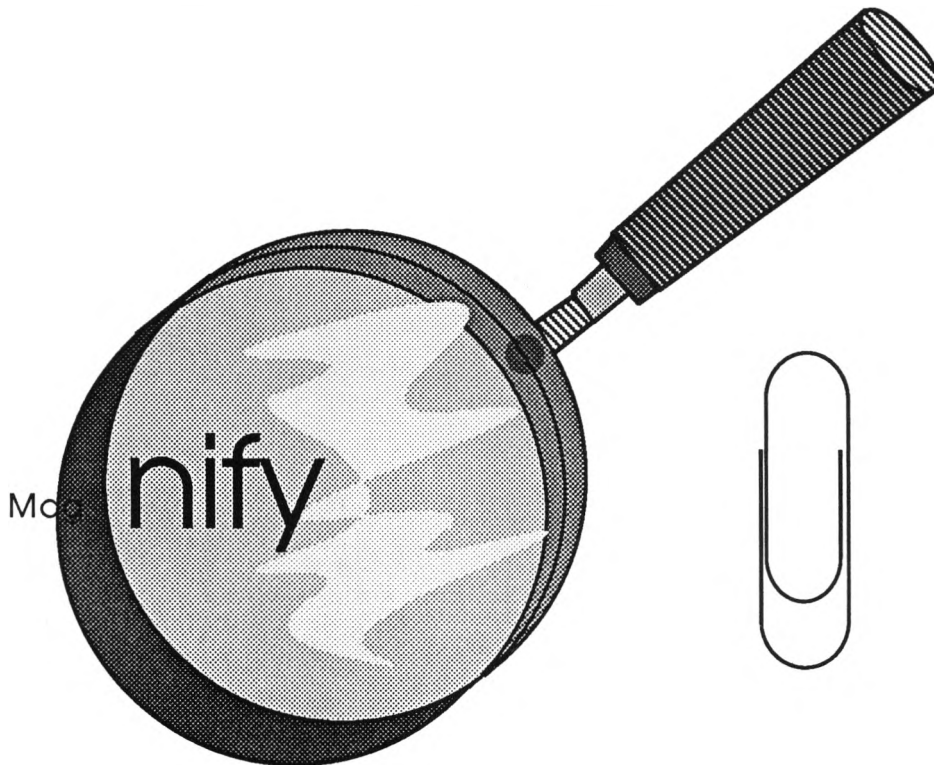
Be brave. Stand up and ask a question. Can't think of a computer question? Well, then ask for advice. Try using this Dear Abbey special, "I'm frustrated by the poor performance I get from a floppy disk and I was wondering if someone here could recommend a reliable, hard one."

They'll be clamoring over the aisles to help you.

* * *

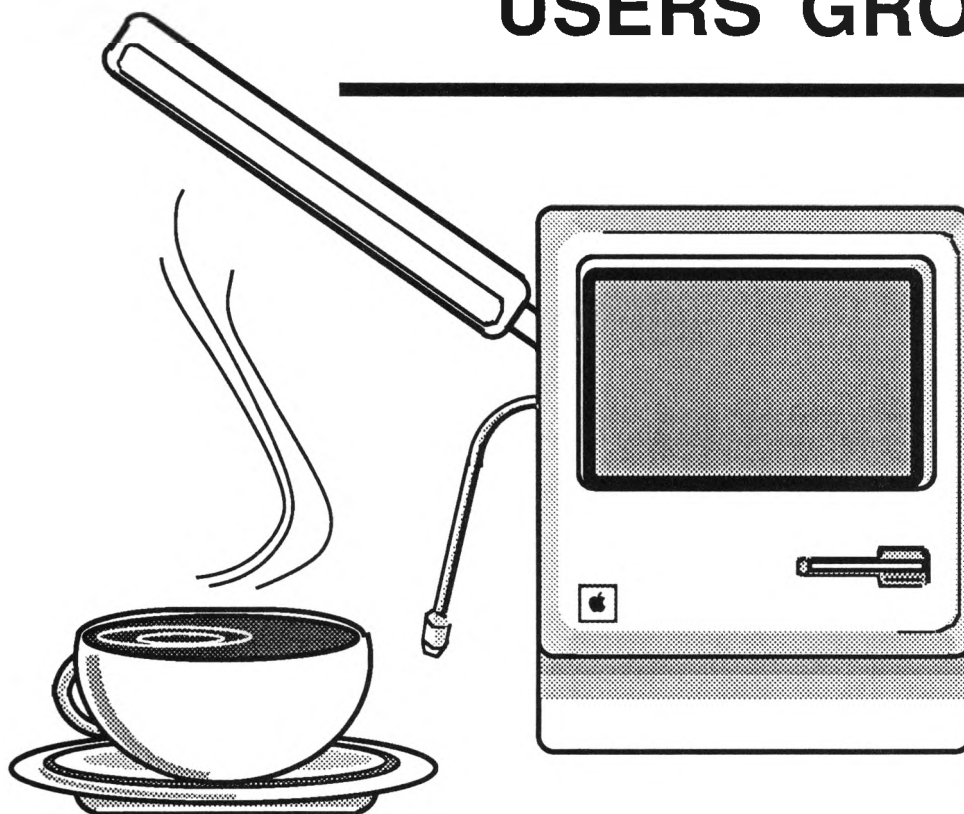
CONFIDENTIAL TO H. C. OF FRISCO: No, John and I don't have a thing going. Quit being so jealous. You're still my one and only. Nobody gives three dots like you...

CONFIDENTIAL TO J. D. OF BERKELEY: Herb is getting suspicious. We have to start being more careful. Thursdays are still on. (Herb thinks I'm at the BMUG meeting.)



Glass, Wire / Wood

NEW! FROM THE _____
**BERKELEY MACINTOSH
USERS GROUP**



The CAP MAC™

PERSONAL COMPUTER AND CAPPUCCINO MACHINE

Features Include:

4 Meg RAM
68020 co-processor
Latte and Double Latte options

Friendly and Very Excited
User Interface
"No Foam" Programmer's Switch

Cappuccino Mac / S. Howard

BMUG
Fall 1986



BMUG Choices

BMUG CHOICES

BMUG Choice Products	389
• The very good, the good, or the only	
The Latest BMUG Disks	402
• Desktop views of recent library additions	
BMUG Newsletter Disk #60	403
• Info on the disk that comes with this newsletter	
BMUG Membership and Product Order Form	405
• Make copies and give to friends	



BMUG Choice Products

We think these products are worth mentioning. Quality ranges from "vastly superior to anything available for any computer" all the way down to "the least objectionable of what's available for the Mac." We've included some items which should be part of a complete office. Order listed does not necessarily indicate order of preference.

Prices indicated are list price / cheapest price we've found. Always assume you can receive a better price somewhere else! Do not assume that a high-priced product is better than one that's almost free. Some public domain software, Shareware, and free products are of higher quality than similar commercial products.

Most products are less expensive when ordered from telephone/mail-order advertisers found in MacWorld, MacUser, or Byte (see examples below). Large computer retail stores, like BusinessLand or Computerland, will probably laugh in your face if you try to bargain down their prices. At least you can try. If they laugh, then you can laugh as you pick up the phone. Most mail order products can be in your hands within two days.

Hard disk drives

Apple HD20

\$1195 / \$750

Apple 20 Megabyte hard disk that fits below the Mac and attaches to the external drive port. By far, the most reliable hard disk drive in the Mac world. Apple Computer, Inc., 20525 Mariani Ave, Cupertino CA 95014, (408)-996-1010

Data Frame 20

\$1095/\$850

External 20 Megabyte drive. Connects to SCSI port on the Mac+. Perhaps not the best value in terms of reliability but has the best support of almost any Mac peripheral. SuperMac Technology, 1901 Old Middlefield Way, Mountain View, 94040, (415)-965-8884

ProAPP 20

\$995 / \$800

A very quiet SCSI drive that can also run on an Apple //. ProAPP, 1475 S. Bascom Ave., Suite 101, Campbell, CA 95008, (408) 559-3552.

WhisperDrive 20

\$795 / \$650

The smallest of the hard disk drives in choice products. One of the few drives that we at BMUG never had break down. Palo Alto MicroSystems. Call or write for address.

Floppy disk drives

Data Space DS-800

\$299 / \$200

We believe that hard disk drives are preferable, but sometimes you need a floppy disk drive and this 800k drive is a good value for your money.

Modems

U.S. Robotics Courier 2400 baud

\$495 / \$380

Another good product from U.S.R. Used on BMUG's BBS. If you can afford it, buy it.



BMUG FALL 1986

Prometheus ProModem 1200 baud **\$549 / \$235**

It has many nice features and is expandable. 100% Hayes compatible. Runs SmartCom fully.

Avatex 1200 **\$129/\$75** **Avatex 1200HC** **\$250/\$107**

A very nice beginner's modem, 90% Hayes compatible. A great value. Includes CompuServe starter kit. Good deals are available for users groups. The Avatex 1200HC is 100% Hayes compatible. MAC (415)-644-0516.

Printers

Apple ImageWriter I (dot matrix) **\$300 / 200**

Yes, you read right. The ImageWriter I is a better printer the ImageWriter II. After being out for longer than a year, Apple still hasn't come out with a reliable successor to the very solid, reliable ImageWriter I, Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014, (408) 996-1010.

Apple LaserWriter Plus (Laser printer) **\$5400 / \$3200**

What do you think of the print quality of this newsletter? (Can you find the pages that weren't printed out on a LaserWriter Plus?)

Network Products (Hardware and Software)

PhoneNET **\$49 / \$34**

Inexpensive AppleTalk clone that works much better than AppleTalk, and uses ordinary phone lines. Farallon Computing, 1442A Walnut #64, Berkeley, CA 94709, (415)-849-2331.

MacServe **\$250**

Software that allows a Mac to serve as a disk server. Works very reliably. Infosphere, Inc., 4730 SW Macadam Ave., Portland, OR 97201, (503)-226-3620

TOPS **\$150 / \$137** (for Mac, more for others)

Complete local area network software for Macintosh, IBM, and DEC VAX computers. (IBMs and DEC VAXen require hardware.) It's a major thrill to be able to manipulate 1-2-3 files with Excel, while they stay on an IBM hard disk. Totally user-transparent and well-done. We used TOPS extensively in preparing this newsletter. Centram Systems West, **

InBox **\$75/\$68**

A great AppleTalk electronic mail system. Easy to use and install. Think Technologies, Inc., 420 Bedford St. Lexington, MA 02173, (617)-863-5595

Maze Wars+ **\$49/\$39**

A new slant on a old game. Up to 32 people can play MazeWars on a AppleTalk network or over a modem. Several mazes and lots of fun. MacroMind, Inc., 1028 W. Wolfram, Chicago, IL 60657, (312)871-0987.



Upgrades

Levco Prodigy 4

From 128K or 512K Macs \$6000

An add-on board which adds M68020 and M68881 chips and four megabytes of memory for the power user. Levco, 6160 Lusk Blvd. # C-203, San Diego, CA 92121, (619) 457-2011. The San Francisco Bay Area dealer is CJS Systems, Berkeley, CA, (415) 849-3730.

Levco One Plus One Upgrade

\$499/\$250

This is Levco's solution for upgrading your Mac Plus to 2 meg. Follows in the Levco tradition of good, clean design. Levco, 6160 Lusk Blvd. # C-203, San Diego, CA 92121, (619) 457-2011. The San Francisco Bay Area dealer is CJS Systems, Berkeley CA 94705 (415) 849-3730.

The MaxPlus

\$499/\$299

The first Mac Plus to 2 meg upgrade. After a wobbly start, they've worked out reliability problems and can be bought just about anywhere. MacMemory, Inc.; 473 Macara Ave., Sunnyvale CA, (408)773-9922

Miscellaneous

LaserServe

\$150

The best of the software laser spoolers. Also spools to ImageWriters. Infosphere, Inc. 4730 SW Macadam Ave, Portland OR 97201, (503)-2263620

Data Space MaxBuffer

\$500

An external print buffer for the ImageWriter. May be useful if you print many documents. Data Space, Markum, Ontario, Canada.

MacTote Carrying Case

\$90 / \$59

The preferred bag for transporting your Mac. Optimum Computer Luggage, 9005 Complex Dr., San Diego, CA 92123, (800) 447-0300.

Sony Microfloppy Disks

Price varies by region

Fuji Microfloppy Disks

Mouse surfaces

\$10 - \$20-\$50

A+ Mouse

\$90 / \$39

An optical mouse that slides very smoothly. Much nicer than Apple's mouse. Mouse Systems Corp, 2600 San Tomas Expy., San Jose CA 95051, (408)-988-0211

Special neoprene pads or plastic trays provide a good traction surface for the mouse. Soft pads eventually wear out but grab best. Hard plastic surfaces can be used on any non-smooth surface (legs, rocks, car seats).

Mouse Pad (hard plastic)

by South Bay Software, P.O. Box 969, Millbrae CA 94030, (415) 579-5455.

One may also buy a large typewriter pad from any office supply store, such as Peninsula Office Supply in the S.F. Bay Area, and cut it to one's desired size.

Digitizers

MacRecorder Plus (voice/analog signal) \$55

The outstanding digitizer designed by Mike Lamoureux. Works with the Mac+. Distributed by BMUG in kit form with terrific software. Now works with the SoundCap™ software from the MacNifty™ folks.

ThunderScan

\$229 / \$169

Has the best resolution, but it's slow. Can digitize only items that can fit into the ImageWriter printer. ThunderWare, Inc., 21 Orinda Way, Orinda, CA 94563, (415) 254-6581.

MacVision

\$400 / \$167

It's fast and it can digitize 3D images from a video camera or videotape. It's resolution isn't quite as good as the ThunderScan. Koala Technologies Corp., 3100 Patrick Henry Drive, Santa Clara, CA 95050, (408) 986-8866

Scan 300

\$2495 / \$1745

A optical scanner that can scan drawings, photographs at 300 dpi. And can do it in just a few seconds. Abaton Technology Corp. 1526 Cloverfield Blvd. Santa Monica CA, 90404, (818)-905-9399

Fans

\$129 / \$79

Fanny Mac. BMUG uses this one on the BBS. Keeps it noticeably cooler. Beck-Tech, 41 Tunnel Rd., Claremont Hotel, Berkeley, CA 94705, (800)-227-2400

Software Products Writing

MacWrite

\$125 / \$70

A classic. Most of the newsletter was composed and printed from MacWrite. Almost everybody has MacWrite. Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014, (408) 996-1010.

HabaWord

\$199 / \$79 (WHEN RELEASED)

An easy-to-use, very powerful word processor. Shows columns on your screen and has MacWrite-like rulers. Haba Systems, 6711 Valjean Ave, Van Nuys, CA 91406

Acta

\$59.95 / \$34

A good outliner, small and fast and a DA too. Supports Undo, unlike most other outline processors. A very useful tool. Symmetry Corp. 761 E University Dr. Suite C, Mesa, AZ 85203, (602)-844-2199

QUED

\$65 / \$49

A great text editor which will eventually become a full featured word processor which should compete with MacWrite, Word, and the other word processing programs soon to be released. Paragon Courseware, 4954 Sun Valley Rd, Del Mar, CA 92104 (619)-481-1477



Spellswell

\$59.95/\$40

Fast and with a large dictionary. Easy to use with a user-modifiable homonym dictionary. The BMUG newsletter was largely spell checked with it. Green, Johnson, Inc., Via Chualar, Monterey CA 93940 (408)-375-2828

Fluent Fonts

\$50 / \$24

Still the best font disks you can buy. Excellent user support. Laser fonts available too. Casady Company, P.O. Box 223779, Carmel, CA 93922, (415) 646-4660.

Fei Ma

\$595 /

Fei Ma is a complete Chinese language word processor and file system. Easy to use, with six methods of entering Chinese characters and all menus, dialog boxes, and the rest in Chinese. Fei Ma c/o Unisource Software Corp., 23 East Street, Cambridge MA 02141.

Allotype Typographics

\$75/\$125/\$40

Wide selection of laser fonts with a science slant. Greek, Polish and chemical fonts are a few of them. Allotype Typographics, 1600 Packard Rd. #5, Ann Arbor MI 48104, (313) 663-1989

Princeton Laser Math Font

\$8 Shareware

Page Layout & Graphics Programs

SuperPaint

\$99/\$79

A program that lives up to its name. A very good paint program and a good MacDraw type program. It can do it all. This is what future program will have to be compared with to make a splash. Silicon Beach Software; PO Box 261430, San Diego CA 92126, (619)695-6956

Paint Cutter

\$39.95 / \$32.95

Part of Accessory Pak 1 from Silicon Beach. Allows full pages for MacPaint files. Silicon Beach, 11212 Dalby PL, San Diego CA 92126, (619)695-6956

Full Paint

\$99 / \$69

Up to 4 open windows can be open at the same time. Rotate, distort and perspective are just a few of the things it can do. Ann Arbor Softworks, 308 1/2 S. State Street, Ann Arbor MI 48104, (313)-996-3838

MacDraw

\$195 / \$132

A powerful layout program for flowcharts, diagrams, art, simple newsletters, etc. A must! Apple Computer, Inc., 20525 Mariani Avenue, Cupertino CA 95014, (408) 996-1010.

GLUE

\$49 / \$25

A revolution in data transfer. Allows one to print to disk almost any file and view it with a viewer or print it or take it into another applications. Solutions, Inc. Box 989, Montpelier VT 05602, (802)-229-0368

CricketDraw

\$200 / \$150 (WHEN RELEASED)

A worthy successor to the old stand-by MacDraw. Will do all and much more than MacDraw. It will love your LaserWriter. Cricket Software, Inc.; 3508 Market St. Philadelphia PA, (215)-387-7955, (800)-662-2444



BMUG FALL 1986

PageMaker 1.2

\$495 / \$369

An excellent program for designing and producing camera-ready page layouts. Recommended for use with hard disk and LaserWriter only. Aldus Corporation, 616 First Avenue, Suite 400, Seattle, WA 98104, (206) 67-8165.

Easy 3D

\$99 / \$82.95

Easy 3D can do three-dimensional solid modeling. Can change light sources and some real neat sculpting tools. The user interface takes some getting used to but is well worth the trouble to learn. Needs a 512 or greater Macintosh. Enabling Technologies, Inc., 600 South Dearborn, Chicago, IL 60605

Databases

OverVUE 2.0

\$295 / \$140

Very fast and simple to use. BMUG uses it for our database, but not for printing labels. ProVUE Development, 222 - 22nd Street, Huntington Beach CA 92648, (714) 969-2431.

FileMaker Plus

\$195 / \$166

A very easy but powerful database. Supports graphics and is well supported. Forethought Inc., 1973 Landings Dr. Mountain View CA 94043 (415)-961-4720, (800)-622-9273

PictureBase

\$69.95 / \$49

A database for graphics. Better than the scrapbook. Your graphic databases can be called from a DA.. Symmetry Corp.; 761 E. University Dr., Mesa AZ 85203; (602)-844-2199, (800)- 624- 2485

Disk Ranger 2.6

\$49/\$39

A disk catalog program. Fast and is HFS compatible. It will also label your disks as well. Mainstay, 28611b Canwood St. Agoura Hills, CA 91301 (818)-991-6540, voice or (818)-991-5037, modem

Languages

MPW (Macintosh Programmers Workshop) Shell \$195, \$195 per language

This is the Developer's product-to-get. It has it all. C, Pascal and Assembler all in a UNIX like shell. Apple Computer, Inc., 20525 Mariani Ave, Cupertino CA 95014, (408)-996-1010

Lightspeed C

\$149 / \$129

An appropriately named, incredibly fast C compiler. Perhaps the best C out there. MacPascal came from these guys. THINK Technologies, Inc., 420 Bedford St., Suite 350, Lexington MA 02173.

Lightspeed Pascal

\$125 / \$87

Lightspeed Pascal is a quiet revolution in programming. It's fast, cheap, and easy to use. The best source-level debugging available for any language on the Mac, and full MacPascal compatibility. THINK Technologies, 420 Bedford Street, Lexington MA 02173

Microsoft Basic

\$150 / \$86

An excellent language for both beginners and experienced programmers. It's up to 3.0 now. Microsoft Corp., 10700 Northrup Way, Bellevue WA 98004, (800)- 426-9400

Microsoft Fortran

\$295

For those people that can't give up bad habits.... Is 77 compatible. Microsoft Corp., 10700 Northrup Way, Bellevue WA 98004, (800)- 426-9400

MacPascal 2.1

\$125 / \$85

Good for learning. Apple Computer, Inc., 20525 Mariani Avenue, Cupertino CA 95014, (408) 996-1010.

TML Pascal

\$99 / \$79

A Pascal compiler for the Mac. Complete. Inexpensive way to get into software development. TML Systems PO Box 361626, Melbourne FL 32936, (305-) 245-1873.

MacNosy 2.28

\$85 / \$59

The best disassembler on the market for the Mac (almost the only one). It can peek at the new ROMs and is HFS compatible. Jasik Designs, 343 Trenton Way, Menlo Park CA 94025, (415)-322-1386.

Calculations/Business

Excel

\$395 / \$224

The most powerful spreadsheet available. There is little reason to consider anything else. Includes macros, and many graph options. Doesn't run in Switcher with MacDraw or MacPaint on a 512 Mac. Microsoft Corporation, 10700 Northrup Way, Bellevue WA 98004, (800) 426-9400.

Stella

\$ / \$89

A very interesting program for modelling systems. Hard to find but worth it. Raves from the people who use it.

Cricket Graph

\$195 / \$125

The best graphing program so far. Accepts data from a wide range of programs. Will do scatter, line, area, bar, column, stacked bar, stacked column and polar. Can make all of these 3-D tools. The user interface takes some getting used to. Cricket Software, Inc.; 3508 Market St. Philadelphia PA, (215)-387-7955, (800)-662-2444

MacSpin

\$79.95 / \$69.95

Amazing stat program. Allows you to see your data in a new way. D2 Software, Inc., 3001 N. Lamar Blvd. #110, Austin TX, (512)-482-8933

Terminal Emulators and Communications Programs

Red RyderUser supported software.\$45 (BMUG disk 2.)

Not just a terminal program— it has every option including clickable macros, definable keyboard options, and a host mode (a mini-language giving one remote service procedures). Authorized users receive Red Ryder Host— a program which allows you to setup your own BBS.



BMUG FALL 1986

MockTerminal (from MockPackage)

User Supported Software (BMUG disk 11.)

A desk accessory. Limited features, but does have automatic redialing so one may use MacWrite, for instance, while calling the BMUG BBS. And the current version (4.34) supports MacBinary. (MockWrite MockPrinter and MockChart are also fantastic.) CE Software, 801 73rd Street, Des Moines, IA 50312, (515) 224-1995.

FreeTermPublic domain software (BMUG disk 2.)

A simple yet useful program. Learn to use your new modem on this package, then "upgrade" to Red Ryder. Has a "Fast Track" command alot like Turbo Download.

UW (Unix Windows)Public domain software (BMUG disk 33.)

Use this program to help you communicate with computers using the UNIX operating system. Very useful for UCB students.

MicroPhone

\$74.95/\$59.95

Slick telecommunications software by Dennis Brothers. Also has scripting option and macro buttons. Easy to use. Wait for version 1.1 Software Ventures Corp., 2907 Claremont Ave., Suite 220, Berkeley CA 94705, (800)-336-6477 or (800)-336-6478 in CA

TurboDownload

\$22

Great downloading utility. This a Desk Accessory that can increase the speed of a Xmodem download from most of the large databases up to 325% at 2400 baud. Mainstay, 28611B Canwood St., Agoura Hills, CA 91301, (818)-991-6540

BackDown

\$30

A desk accessory that allows you to download programs across your modem and telephone line while you use your Mac to do other things. You can even quit applications and return to the Finder; BackDown will continue working. Well worth the shareware fee. Ron Risley, PO Box 1724, Orange CA 92668

Smartcom II

\$149/\$89

Easy for the novice, powerful enough for the person who uses it every day. One of the most intuitive telecom program for the Mac. Hayes Microcomputer Products, Inc, PO box 105203, Atlanta, GA 30348, (404)449-8791

Utilities

General utilities on BMUG disk 60

Public Domain/User Supported/Free Software

The disk contains Switcher, RAM Start, Fedit, Resource Editor, Key Caps, Disk Info, Extras. This is the disk you'd want if you were stranded on a remote island.

Servant

\$BMUG update disks

A combination Finder, Switcher, ResEdit that, in final release, will make life with the Macintosh better. For now, author Andy Hertzfeld provides free early versions via users groups and modem services. Bugs persist (Thanksgiving 1986 version is .89) yet this program's power is incredible. Super fast and efficient finder, amazing click-on-window switcher, and icon-based ResEdit. Many more features like scripting not yet implemented.

Multi-Scrap

\$BMUG disk 20.9

Allows you to have several scrapbooks on your disks. Is HFS compatible.



LaserPrint DA

\$BMUG Disk 33.1

A very useful accessory if you use a LaserWriter. Downloads PostScript files to the LaserWriter in the background. Gives you back control much faster.

Copy II Mac

\$40 / \$19

For making an archival backup of copy-protected software that you've PURCHASED. File recovery and modification software. An excellent buy except that the third program on the disk, MacTools, isn't very useful with HFS. Central Point Software, 9700 S.W. Capitol Highway, #100, Portland, OR 97219, (503) 244-5782.

MacBackup

\$59.95/\$40

This will backup the few things that Copy II Mac can't and has one of the best icons. Smarter than Copy II Mac. ROC Software (607)-257-0043

Hard Disk Utility

\$90 / \$72

Puts PURCHASED programs on a hard disk. (Ask them what the initials stand for.)

HFS Locator

\$39/22

Great tool for hard disk drives. Will quickly search for any file. It can also do a great many other things, like copy, launch, move, make a catalog. Nice. **Five Stars**

DiskExpress

\$29/\$19

Another great tool for hard disk drives. DiskExpress will defragment your hard disk which should speed it up as well as save some wear and tear on your hard disk drive. ALSoft, Inc., PO Box 927, Spring, TX 7738

Miscellaneous

Dark Castle

\$49 / \$25

The game of year. It has something for every one and will keep you at it for weeks. Great animation and graphics. Silicon Beach, 11212 Dalby PL, San Diego CA 92126 ,(619)695-6956

Puppy Love

\$19.95 / \$11.95

Puppy Love is a program that has to be seen to be believed. Perhaps the easiest way to learn about programming and lots of fun as well. Addison-Wesley Publishing Co, Inc., Jacob Way, Reading MA, 01867

Quik Circuit

\$99 /

Printed circuit board layout program. One may submit a data disk or transmit layout by modem and receive PC board back in a week. Douglas Electronics, Inc., San Leandro.

MacProject

\$125 / \$85

Critical path method analysis for project management. Includes resource and personnel allocation. A must for project managers. Apple Computer, Inc., 20525 Mariani Avenue, Cupertino CA 95014, (408) 996-1010.

LabView

\$1999

LabView turns your Mac into a virtual machine capable of controlling and monitoring standard IEEE-488 instruments. Acquires, massages, and reports data. LabView is unique because it uses a fully iconic programming language that is easy to use and debug. National Instruments, 12019 Technology Blvd., Austin TX 78727-6204.

Smart Alarms

\$49/\$27

The best reminder program out there. After setting it from a DA, it will remind you of your appointments, birthdays, holidays with a gentle beep (or a not so gentle beep). Imagine Software. 2000 Center Street, #1260, Berkeley CA 94704 (415)-769-4033

Magazines

MacWorld

/ \$24 per year

Monthly. Once again becoming the Mac magazine to have. Under the pressure of competition, MacWorld has made great strides in content and substance. PC World (415) 861-3861. Subscription service: P.O. Box 54529, Boulder CO 80322-4529, (800) 525-0643.

MacUser

/ \$22 per year

Monthly. Good advertisements and good content. Still slipping, needs to be shaken up abit. Great sidebar articles. MacUser Publications, Inc., 25 West 39th Street, New York NY 10018, (212) 302-2626 or (415) 382-8500.

MACazine

/ \$18 per year

Monthly, like MacWorld — but friendlier, more up-to-date. ICON CONCEPTS CORPORATION, The MACazine, PO Box 1936, Athens TX 75751.

MacTutor

/ \$24 per year

Monthly, good for programmers. MacTutor, PO Box 846, Placentia CA 92670, (714) 993-9939.

InfoWorld

/ \$31 per year

Weekly, the latest computer news. Not quite the same without Dvorak, though. InfoWorld, CW Communications, Inc., 1060 Marsh Road, Suite., C-200, Menlo Park CA 94025, (800) 328-4602. Subscriptions, (800) 544-3712.

Byte

\$21 per year

Monthly. Good technical source and general computing theory. BYTE Subscription Dept., PO Box 597, Marysville NJ 08836-9956.

Computer Currents

Free in the San Francisco Bay area

Twice per month, good combination of computer news, rumors, ads, and articles. Computer Currents, Center Productions, 2550 - 9th Street, Berkeley CA 94710, (415) 848-6860.

The Mouseketeer

Free, when found, and well worth the price!

Published occasionally by the unfamous, perhaps infamous, commentator Alfonso Qwerty. Find it at your local user group. Be on the lookout for something new from Mr. Qwerty.

MacInTouch

\$48, US, \$58, Canada, \$85, rest of the world

A useful informative monthly magazine. Accepts no advertising. They always have some of the best reviews of both software and hardware. More technical than MacUser but less technical than MacTutor. PO Box 786, Framingham MA 06707.



Books

The Mouse Droppings Book of Macintosh Hints, or...

What Apple Didn't Tell You About Your Macintosh. \$8.95

This is perhaps the best book to get hints and tips from. Also has the best index of just about any Macintosh book. Compiled by Philip Russell, this is the one to get. Philip Russell, 430 SW Crest Circle, Walport OR 97394.

Mac Insights

\$18.95/\$14.95

This should be one of the first books that a Mac owner gets. Clever and well written with another good index. Written by Lon Poole. Microsoft Press.

Using the Mac Toolbox With C

\$23.95/\$18.95

Excellent reference for programming and learning C. Fred H., Jim, and Dave. Sybex.

Inside Mac

\$69/ \$22.50/\$19.95

The definitive book about programming on the Macintosh. A must for all programmers of Macintosh software. Addison-Wesley.

Microsoft Word Made Easy

\$15

Macintosh Edition, Hoffman, Osborne / McGraw Hill Books.

Macintosh Pascal Illustrated,

The Fear and Loathing Guide

\$17

Wonderful book to read or even to learn Pascal. Find out about love, Pascal and the Toolbox all in one book. Still the best book on learning how to program for beginners. Scott Kronick, Addison - Wesley, 1985.

PostScript Language, Reference Manual,

Volumes 1 and 2

\$23

Detailed information on the LaserWriter's programming language. Adobe Systems, Inc., Addison - Wesley, 1985.

Mail/Telephone-Order Vendors

MacConnection

Reliable service (!) and consistently good prices. 14 Mill St., Marlow NH 03436. (800-622-5472)

ComputerWare

Good prices. You can try out the software without any hassles. Mail order too. 109 California Ave., Palo Alto, CA 94306, (415-323-7557) in CA (800-323-1133); rest of the world (800-235-1155).

Software For Less

Nice store with nice people and a large selection of software to try. Software For Less, 950 Taraval, San Francisco CA 94116, (415)-753-1066.



BMUG FALL 1986

MAC

Great prices and just starting to go into mail order. (415)-644-0516

Best Macintosh BBS

Mac 9

The best Macintosh BBS doesn't even run on a Macintosh. It's a FIDO board that runs on one of those other computers. A complete Macintosh download section and a lively message area with quite a few Developer's make this the one to know. It's also one of the few Official Apple Developer's BBS's. The SYSOPs are knowledgeable about all aspects of the Macintosh and are friendly as well. A Macintosh version of FIDO is in the works so log on and see. 300, 1200, and 2400 baud. (415)-665-7373 or 753-3002.

Best Macintosh Newsletter (other than this one, of course)

Mouse Tracks

\$24(newsletter only)

\$36(full membership)

There must be something in the air in OR. At times, the three major Mac Magazines have tips and hints mainly from this newsletter and the Mouse Droppings newsletter. Great articles explaining in easy terms some of the harder things on the Mac. It might be worth getting this than one of the mainstream Macintosh magazines. Portland Macintosh User Group, Inc., PO Box 8895, Portland OR 97207.

Best Small Macintosh Newsletter

Mouse Droppings

\$12(newsletter only)

\$24(full membership)

For a second time in a row, this is by far the best small newsletter for any computer. It is in a class by itself. It's full of hints, how to articles and just general good sense. Mail to Macintosh User Group of Corvallis, 520 NW Oak St., Corvallis OR 97394.

Best Speaker

Marc Canter

\$Pizza

The leader of MacroMind. Truly is a great fun speaker. He knows a little bit of everything and tells it like it is.

BMUG Controversial Choices

The Bookman Font.

\$6195/\$195 (comes with LaserWriter Plus)

Tell us what you think about the font that the newsletter is printed in. Some people like it and others don't (why else would it be in controversial choice products?) Here's your chance, what font would you like the next newsletter printed in? Adobe Systems, Inc., 1870 Embarcadero RD., Palo Alto CA 94303

Msg. #2817 in **Public Messages** Posted on 12/03/86 at 20:40:15

To: ALL From: LINDA CUSTER

Subject: Micah Woes

I had a very long chat with Bob Miller of Solitare, the new owners of Micah, this afternoon. Basically, the BMUG newsletter goes to press this week (to be out by MW Expo), and we were considering what to say about Micah in our choice products section.

First, Micah's new address is 1201 San Louis Obispo Avenue, Hayward CA. Their phone number can be easily found by asking directory assistance for Hayward CA (area code 415) and asking for Solitare. The number is (415) 487-8300. However, they are not taking support calls at all for at least a few weeks, as they just don't have the technical staff in place to do that yet.

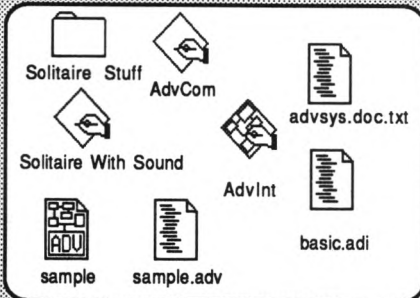
They've moved from their old address into what is more of a warehouse/manufacturing facility, and plan on doing Micah manufacturing in-house now. (Micah used to contract out for mfg.) They're not shipping much product at all yet, mostly just very old backorders. They are not able to provide any technical support and can't honor warranty repairs for the time being. They legally have to consider anyone with warranty problems to be a Micah creditor, and creditors have to get in line. Even if you buy a Micah today, new, you will still have some trouble getting warranty repairs, although you'll be higher up on the priority list than those who bought Micahs before Chapter 11.

Hopefully, technical support and full service for all Micahs will resume around the first of the year ("in a few weeks"). Micah's main problem right now is that the drive manufacturer is taking advantage of the Chapter 11 situation to refuse taking back defective drives. So Micah can't get drives swapped.

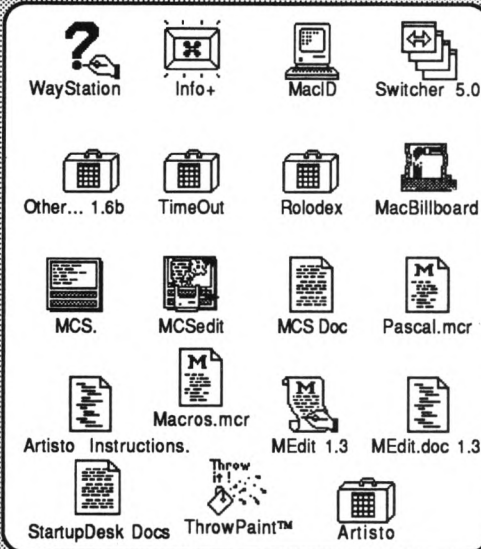
Tony, the former president, is not involved with Micah any more. Solitare is in the process of building a new staff. They intend to emerge from Chapter 11 and hold onto Micah, they still plan on offering 30 meg upgrades to current internal Micahs, and they plan to offer a 30 meg external drive as well as larger drives with tape backup units. When? That's a good question.

The Latest BMUG Disks

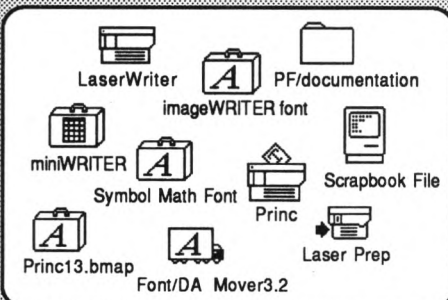
Here are the highlights of the latest disks in our library. This is by far the cheapest way to get quality software! Please support BMUG by buying our disks...



Games Disk #38



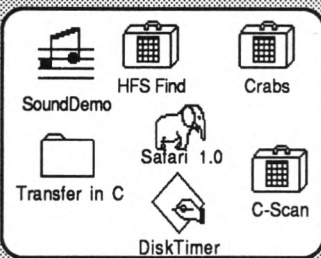
Fall '85 Companion Disk #40



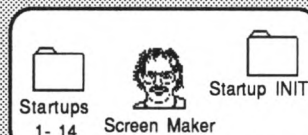
Princeton Laser Font Disk #42



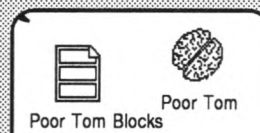
BMUG Disk Catalog Disk #43



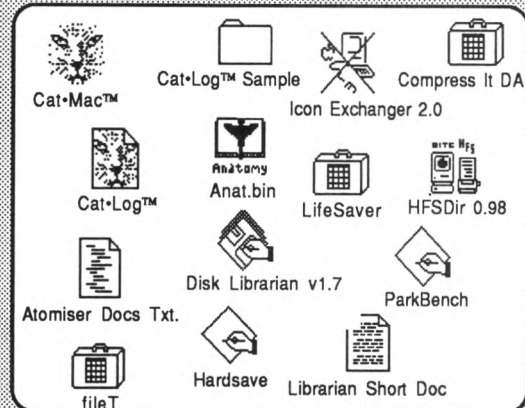
Programming Disk #45



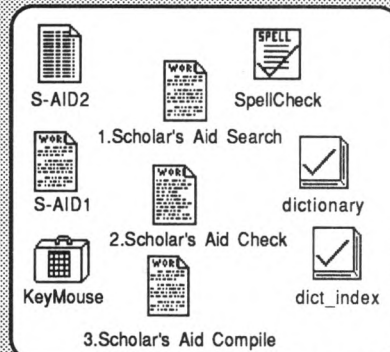
Startup Screens Disk #46



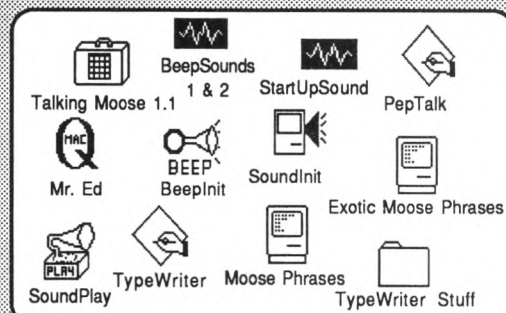
Poor Tom Disk #44



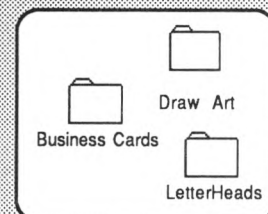
Utilities Disk #39



Spell Checker Disk #41



Sound Program Disk #47



PageMaker Template Disk #48



BMUG FALL 1986

BMUG Fall 1986 Membership Disk



TalkAlarm



Smallview



Layout



DeskZap



FreeTerm 1.8



Oasis



LabelPrinter



MacSnoop™



CrystalRaider



TalkAlarm

Have you ever wanted your Mac to gently wake you up in the morning? Well TalkAlarm isn't going to do it but it's a start. (Be sure and put MacinTalk in your System folder if you want it to talk).



Layout

Layout is a great utility to make your desktop look the way you want. It will allow you to customise the way your Finder displays the default font type and size as well as many other defaults such as icon spacing. Be careful: only use on disks that are backed up.



DeskZap

DeskZap is a SetFile-like DA that has a utility for printing (i.e. File Diddler) and telecommunications.

Shareware \$10



FreeTerm 1.8

FreeTerm is a simple, solid terminal program. It has a Fast-Track feature which can speed up downloads.



LabelPrinter

A nice, simple labeling program. It can do addresses too. Do not use for large-scale operations as it can be hard to configure.



Crystal Raiders

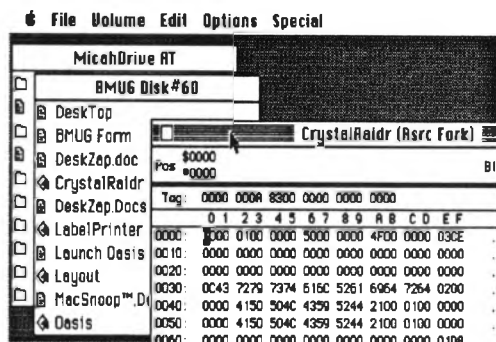
A fun game in which the object is to capture crystal balls that are guarded by shooting enemies. A fast-paced game that will keep you on your toes.



The Animator

The program and resource code (in LightSpeed Pascal) of the program "Animator" as described in the newsletter article beginning on page 78.

BMUG FALL 1986



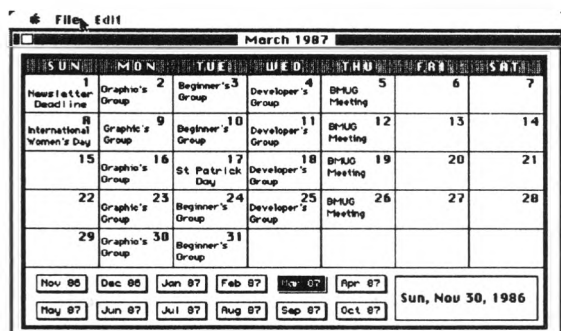
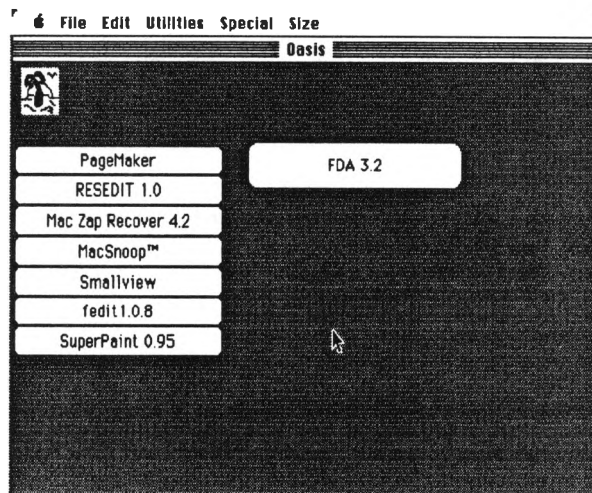
MacSnoop is a file editor somewhat like FEdit but with a user interface like ResEdit. You can open several files at once. Very nice. Look for future enhancements.

ShareWare \$25



Oasis might get you to forget that Apple ever gave you a Finder. It's the best mini-finder out. Fast with lots of utilities. It can do just about everything you'd want a Finder to do. Unfortunately, it only works with HFS systems.

ShareWare \$15



Smallview is a great little calendar program. It will print out calendars to the year 2038. Good for appointments and holidays.

ShareWare \$12

Summary of the BMUG disk library

The most complete and up-to-date disk library available anywhere, not full of redundant copies of system and finder.

1. **Fall '84 companion disk:** File, Rolodex, games, 3D demo, Font Editor, Hex Dump, Screen Maker, MS-BASIC programs.
2. **Communications disk:** BinHex 4.0 & 5.0, Red Ryder 9.4, Freeterm, Versaterm, Packit, list of BBS's
3. **Utilities:** Banner, Alert/Dialog Editor, XLisP, Rmover, RAM disks, DA's: SetFile, Calendar, etc.
4. **FORTH disk:** examples, utilities, games (needs MacForth level 1)
5. **FileVision disk:** templates (requires FileVision)
6. **Font making disk:** Font Editor, doubler, lister, mover, tutorial, new key caps DA, font utilities
7. **Font library disk:** lots of text, technical, scientific fonts
8. **Font library disk:** banner/sign/display fonts
9. **Graphics disk:** dozens of programs to show off your Mac! Fractals programs, PepsiCaps, Trench, 3-D stuff, & MacPaint utils.
10. **Spring '85 companion disk:** SetFile DA, AutoDial, Binhex, MockPackage, ResEdit, FreeTerm, FEdit, Edit, Font/DA Mover
11. **Desk Accessory Galore disk:** 50 Desk accessories of all kinds.
12. **MacRecorder II disk** (comes with kit). Recorder/player, etc. (Can play and edit w/o hardware)
13. **Games disk:** Altered Finder, Amazing, Smoothtalker demo, Concentration, Hangman, Mouseball, Reversi, etc.
14. **Utilities & games:** Clocks, BashBB, Megaroids, PREC's (new ImageWriter paper size templates), Triva Savant demo, Quick Utilities, Missile Command, etc.
15. **LaserWriter fonts & installation,** latest Switcher (4.8 or greater. Requires at least 512K)
16. **Official system/finder update** (sys 3.2, finder 5.3: May 1986).
17. **Various:** 512K Copy, BigBen DA, Boston Font, Word Count, etc.
18. **Developers' SIG disk:** "C" tutorials, source code for exams in Sp. '85 Developers' book, REdit
19. **Various:** DA's (Freemem, VT52, Dvorak), Communique, Icon Editor, Wizard's Fire, Fonts
20. **Fall '85 companion disk:** ResEdit, Customize MacDraw, Switcher, Paint Copier, DA sampler, MockPackage, FreeTerm, FEdit
21. **Games** (bomber, core wars, maze), graphics, star chart (BASIC), font librarian, font tester, fonts.
22. **ShareWare M68000 Cyclan assembler** (requires 512K Mac)
23. **Miscellany:** Paint copier, Quick File; Way station, Kermit term.
24. **Educational software** (various levels)
25. **AppleTalk/PhoneNET** local area network utilities, examples, games. Available late 1986.
26. **Games:** Dungeons of Doom, MacLuff, ThinkAhead+, Brickles, etc.
27. **Games:** Billiards, Space War, Invaders, etc.
28. Lots more DA's, plus **Calculator Construction kit demo**
29. **Utilities:** Icon Maker, Draw to Paint conv, Read MW, FKEYs
30. **Excel and Multiplan templates:** taxes, Airways, Telcon, real estate, etc. Most require Excel.
31. **UNIX™ utilities:** Unix Windows, Versaterm 1.0, MockPackage, VT52 DA, Packit, Binhex, file filter, edit, etc.
32. **Graphics stuff:** Dynamo, Grey fox, draw templates, 3-D stuff, Fido
33. **Printer utilities** for ImageWriter & LaserWriter
34. **Games:** Pinball Construct Set Demo, Cap'n Magneto
35. **Programmers:** Softworks BASIC, ADA demo, state machine
36. **Music:** Concertware templates & Jam Session
37. **Basic programs** (requires MS-BASIC 2.0)
38. **Solitaire, Adventure Game maker**
39. **Utilities:** Icon exchanger, anatomy, hard disk utilities, disk cataloggers, DA's: compress it, lifesaver, file T
40. **Spring '86 companion disk**
41. **Spell Checker** and scholar's aide (requires MS file, Word, and BASIC), keymouse DA (to type with mouse)
42. **Princeton Laser Font** and miniWriter
43. **BMUG Disk Catalog & cataloggers**
44. **Poor Tom** (database of words)
45. **Programming disk** (C programming examples)
46. **Startup Screens, backdrop, Startup INIT**
47. **Sound Programs:** Beep Sound, Mr. Ed, Sound Play, Pep Talk, etc.
48. **PageMaker templates**
- M. **Modula -2 compiler** from ETH/Zurich (set of 3 disks: System/program, doc, and library files)

Summary of the BMUG back issues

BMUG's "newletters" - each over 200 pages packed with tips, tricks, reviews, advice, and NO ADVERTISING. Timeless info for the Mac user. Each includes a disk.

Spring '85 : (only 120 pages) - Disk survey, telecommunications, lots of software reviews

Fall '85: Hardware, application-oriented tutorials, and a good deal of general information and reference material

Spring '86: Communications, Graphic Design, Hacking, Programming, and various reviews.

Developer Spring '85': A simple and easy to read tutorial on writing Macintosh applications in C. By the authors of "Using the Mac Toolbox with C."

Order form for BMUG products as of 12/86. Thanks for supporting BMUG!

Contents, , availability and pricing subject to change without notice. Please allow up to 4 weeks for delivery. © 1986 BMUG, inc.

Software disks: (All are Macintosh single sided. We update our disks VERY often!)

Please indicate which disks you want. See page 405 for listing.

		@ \$3.00 ea =	
----------------------	-------	---------------	----------------------

Disk Subscription: the next 10 of our newest disks mailed as soon as they're published starting with the number you specify..... Start with BMUG disk #

		@ \$30.00 =	
----------------------	-------	-------------	----------------------

Cables:

☐ Keyboard extension cable	12' long, replaces the Mac keyboard cable		@ \$7.00 ea =	
---	---	-------	---------------	----------------------

☐ Mini DIN 8 to DB-9	☐ Mini DIN 8 to mini DIN 8.		@ \$10.00 ea =	
(adapter)	(10' extension)			

Hardware Kits:

MacRecorder II Plus:	an A/D converter and MacNifty™ - compatible		@ \$55.00 ea =	
	speech digitizer			

BMUGNET Kit (Hacker's version, DB9 assembled only - limited stock)		@ \$20.00 ea =	
--	-------	----------------	----------------------

Back Issues:

See page 405
for more details

Spring '86 & disk #40	☐	Spring '85 & disk #10	☐	Fall '86 & disk #60 (over 400 pages)	☐	@ \$18 ea. or @ \$20 ea. (for first class)	
-----------------------	--------------------------	-----------------------	--------------------------	--------------------------------------	--------------------------	--	----------------------

Fall '85 & disk #20	☐	Spring Developer '85 & disk #18	☐	@ \$15 ea. or @ \$18 ea. (person or book rate) (first class)	
---------------------	--------------------------	---------------------------------	--------------------------	---	----------------------

Membership:

Note: Newsletters are usually published late in the semester, so please be patient.

Each semester's membership includes a disk and Newsletter, use of the BBS and Helpline, and more.

By the BMUG calendar, Spring is Feb. to July, and Fall is July to Feb. As soon we receive your payment we will send you a confirmation. Please help us by joining 2 semesters at a time.

Spring '87	☐	Fall '87	☐	Spring '88	☐	@ \$40.00 /yr. or @ \$20.00 ea. =	
------------	--------------------------	----------	--------------------------	------------	--------------------------	-----------------------------------	----------------------

All prices include domestic postage. For foreign postage, add to each of the following:

Newsletter/Kit/Membership..\$3 for Can. and Mex., \$10 for overseas.

Disk/ Cable..\$.50 for Can. and Mex., \$1 for overseas.

Foreign postage =	
-------------------	----------------------

Name: _____ Date: _____

Donation = (tax deductible)	
-----------------------------	----------------------

Address: _____

Make check payable to "BMUG" with U.S. negotiable funds.

\$	
----	----------------------

City: _____ State: _____ Zip Code: _____

Renewal?	☐
----------	--------------------------

TOTAL ENCLOSED

Country: _____ Area Code & Phone: _____

Change of Address?	☐
--------------------	--------------------------

BMUG, Inc. 1442A Walnut St. #62 Berkeley CA 94709 (415) 849-9114

Summary of the BMUG disk library

The most complete and up-to-date disk library available anywhere, not full of redundant copies of system and finder.

1. **Fall '84 companion disk:** File, Rolodex, games, 3D demo, Font Editor, Hex Dump, Screen Maker, MS-BASIC programs.
 2. **Communications disk:** BinHex 4.0 & 5.0, Red Ryder 9.4, Freeterm, Versaterm, Packit, list of BBS's
 3. **Utilities:** Banner, Alert/Dialog Editor, XLisP, Rmover, RAM disks, DA's: SetFile, Calendar, etc.
 4. **FORTH disk:** examples, utilities, games (needs MacForth level 1)
 5. **FileVision disk:** templates (requires FileVision)
 6. **Font making disk:** Font Editor, doubler, lister, mover, tutorial, new key caps DA, font utilities
 7. **Font library disk:** lots of text, technical, scientific fonts
 8. **Font library disk:** banner/sign/display fonts
 9. **Graphics disk:** dozens of programs to show off your Mac! Fractals programs, PepsiCaps, Trench, 3-D stuff, & MacPaint utils.
 10. **Spring '85 companion disk:** SetFile DA, AutoDial, Binhex, MockPackage, ResEdit, FreeTerm, FEdit, Edit, Font/DA Mover
 11. **Desk Accessory Galore disk:** 50 Desk accessories of all kinds.
 12. **MacRecorder II disk** (comes with kit). Recorder/player, etc. (Can play and edit w/o hardware)
 13. **Games disk:** Altered Finder, Amazing, Smoothtalker demo, Concentration, Hangman, Mouseball, Reversi, etc.
 14. **Utilities & games:** Clocks, BashBB, Megaroids, PREC's (new ImageWriter paper size templates), Triva Savant demo, Quick Utilities, Missile Command, etc.
 15. **LaserWriter fonts & installation,** latest Switcher (4.8 or greater. Requires at least 512K)
 16. **Official system/finder update** (sys 3.2, finder 5.3: May 1986).
 17. **Various:** 512K Copy, BigBen DA, Boston Font, Word Count, etc.
 18. **Developers' SIG disk:** "C" tutorials, source code for exams in Sp. '85 Developers' book, REdit
 19. **Various:** DA's (Freemem, VT52, Dvorak), Communique, Icon Editor, Wizard's Fire, Fonts
 20. **Fall '85 companion disk:** ResEdit, Customize MacDraw, Switcher, Paint Copier, DA sampler, MockPackage, FreeTerm, FEdit
 21. **Games** (bomber, core wars, maze), graphics, star chart (BASIC), font librarian, font tester, fonts.
 22. **ShareWare M68000 Cyclan assembler** (requires 512K Mac)
 23. **Miscellany:** Paint copier, Quick File; Way station, Kermit term.
 24. **Educational software** (various levels)
 25. **AppleTalk/PhoneNET** local area network utilities, examples, games. Available late 1986.
 26. **Games:** Dungeons of Doom, MacLuff, ThinkAhead+, Brickles, etc.
 27. **Games:** Billiards, Space War, Invaders, etc.
 28. Lots more DA's, plus **Calculator Construction kit demo**
 29. **Utilities:** Icon Maker, Draw to Paint conv, Read MW, FKEYs
 30. **Excel and Multiplan templates:** taxes, Airways, Telcon, real estate, etc. Most require Excel.
 31. **UNIX™ utilities:** Unix Windows, Versaterm 1.0, MockPackage, VT52 DA, Packit, Binhex, file filter, edit, etc.
 32. **Graphics stuff:** Dynamo, Grey fox, draw templates, 3-D stuff, Fido
 33. **Printer utilities** for ImageWriter & LaserWriter
 34. **Games:** Pinball Construct Set Demo, Cap'n Magneto
 35. **Programmers:** Softworks BASIC, ADA demo, state machine
 36. **Music:** Concertware templates & Jam Session
 37. **Basic programs** (requires MS-BASIC 2.0)
 38. **Soltaire, Adventure Game maker**
 39. **Utilities:** Icon exchanger, anatomy, hard disk utilities, disk catalogers, DA's: compress it, lifesaver, file T
 40. **Spring '86 companion disk**
 41. **Spell Checker** and scholar's aide (requires MS file, Word, and BASIC), keymouse DA (to type with mouse)
 42. **Princeton Laser Font** and miniWriter
 43. **BMUG Disk Catalog & catalogers**
 44. **Poor Tom** (database of words)
 45. **Programming disk** (C programming examples)
 46. **Startup Screens, backdrop, Startup INIT**
 47. **Sound Programs:** Beep Sound, Mr. Ed, Sound Play, Pep Talk, etc.
 48. **PageMaker templates**
- M. **Modula -2 compiler** from ETH/Zurich (set of 3 disks: System/program, doc, and library files)

Summary of the BMUG back issues

BMUG's "newletters" - each over 200 pages packed with tips, tricks, reviews, advice, and NO ADVERTISING. Timeless info for the Mac user. Each includes a disk.

Spring '85 : (only 120 pages) - Disk survey, telecommunications, lots of software reviews

Fall '85: Hardware, application-oriented tutorials, and a good deal of general information and reference material

Spring '86: Communications, Graphic Design, Hacking, Programming, and various reviews.

Developer Spring 85': A simple and easy to read tutorial on writing Macintosh applications in C. By the authors of "Using the Mac Toolbox with C."

Order form for BMUG products as of 12/86. Thanks for supporting BMUG!

Contents, , availability and pricing subject to change without notice. Please allow up to 4 weeks for delivery. © 1986 BMUG, inc.

Software disks: (All are Macintosh single sided. We update our disks VERY often!)

Please indicate which disks you want. See page 405 for listing.

..... @ \$3.00 ea =

Disk Subscription: the next 10 of our newest disks mailed as soon as they're published starting with the number you specify..... Start with BMUG disk #

..... @ \$30.00 =

Cables:

☐ Keyboard extension cable 12' long, replaces the Mac keyboard cable

..... @ \$7.00 ea =

☐ Mini DIN 8 to DB-9 (adapter) ☐ Mini DIN 8 to mini DIN 8. (10' extension)

..... @ \$10.00 ea =

Hardware Kits:

MacRecorder II Plus: an A/D converter and MacNifty™ - compatible speech digitizer

..... @ \$55.00 ea =

BMUGNET Kit (Hacker's version, DB9 assembled only - limited stock)

..... @ \$20.00 ea =

Back Issues:

See page 405 for more details

Spring '86 & disk #40

☐

Spring '85 & disk #10

☐

Fall '86 & disk #60 (over 400 pages)

☐

@ \$18 ea. or @ \$20 ea. (for first class)

Fall '85 & disk #20

☐

Spring Developer '85 & disk #18

☐

@ \$15 ea. or @ \$18 ea. (person or book rate) (first class)

Membership:

Note: Newsletters are usually published late in the semester, so please be patient.

Each semester's membership includes a disk and Newsletter, use of the BBS and Helpline, and more. By the BMUG calendar, Spring is Feb. to July, and Fall is July to Feb. As soon we receive your payment we will send you a confirmation. Please help us by joining 2 semesters at a time.

Spring '87

☐

Fall '87

☐

Spring '88

☐

@ \$40.00 /yr. or @ \$20.00 ea. =

All prices include domestic postage. For foreign postage, add to each of the following:

Newsletter/Kit/Membership..\$3 for Can. and Mex., \$10 for overseas.

Disk/ Cable..\$.50 for Can. and Mex., \$1 for overseas.

Foreign postage =

Name: _____ Date: _____

Donation =

(tax deductible)

Address: _____

Make check payable to "BMUG" with U.S. negotiable funds.

City: _____ State: _____ Zip Code: _____

TOTAL ENCLOSED

Renewal?

☐

Country: _____ Area Code & Phone: _____

Change of Address?

☐

BMUG, Inc. 1442A Walnut St. #62 Berkeley CA 94709 (415) 849-9114



with apologies to Berke Breathed